HREC-5712-2
LMSC-HREC D225632

LOCKHEED MISSILES & SPACE COMPANY
HUNTSVILLE RESEARCH & ENGINEERING CENTER
HUNTSVILLE RESEARCH PARK
4800 BRADFORD DRIVE, HUNTSVILLE, ALABAMA

STUDY ON PROPELLANT DYNAMICS DURING DOCKING

FINAL REPORT

March 1972

Contract NAS8-25712

Prepared for National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama 35812

by

G.C. Feng
S. J. Robertson

APPROVED: B. Hobson Shirley

B. Hobson Shirley, Supervisor
Aerophysics Section

George D. Reny

George D. Reny, Manager
Aeromechanics Dept.

J.S. Farrior

J.S. Farrior
Resident Director

FOREWORD

This document is the final report of a research program performed by Lockheed Missiles & Space Company, Inc., Huntsville Research & Engineering Center, while under contract to NASA-Marshall Space Flight Center, Contract NAS8-25712. The report summarizes a research effort accomplished between 29 June 1971 and 15 March 1972. Tasks completed before this period were reported in an interim report, LMSC-HREC D225157, dated June 1971. The NASA technical monitor of the contract was Mr. Frank Bugg, S&E-AERO-DDS.

SUMMARY

The marker-and-cell numerical technique was applied to ~~the study~~ of axisymmetric and two-dimensional flow of liquid in containers under low gravity conditions. The purpose of the study was to provide the capability for numerically simulating liquid propellant motion in partially filled containers during a docking maneuver in orbit. A computer program to provide this capability for axisymmetric and two-dimensional flow was completed and computations were made for a number of hypothetical flow conditions.

In order to extend the numerical simulation capability to more realistic flow conditions, a research effort was undertaken to develop a three-dimensional marker-and-cell computational technique. For this initial effort container boundaries were limited to rectangular shapes. A pilot computer program was successfully developed as a result of this research effort. The computer program requires 64K core storage with four drum areas for temporary storage. Computations were made for several test cases with reasonable results obtained. This pilot program can be more fully developed to include such features as the capability for treating containers with curved boundaries.

CONTENTS

Section		Page
	FOREWORD	ii
	SUMMARY	iii
1	INTRODUCTION	1
2	FORMULATION	
	2.1 Governing Equations	2
	2.2 Finite Difference Formulation	4
3	NUMERICAL SCHEMES	8
	3.1 The MAC ~~Com~~puting Technique	8
	3.2 Boundary Conditions	11
	3.3 Displacing of Marker Particles	15
	3.4 Forces and Moments	15
4	THE COMPUTER PROGRAMS	17
	4.1 User's Guide of the LHMAC2 Program	17
	4.2 User's Guide of the LHMAC3 Program	20
5	EXAMPLES	25
6	CONCLUSIONS AND RECOMMENDATIONS	27
	REFERENCES	28

Section 1
INTRODUCTION

The Space Shuttle and Space Station programs as currently planned have generated much interest in recent years in the dynamic behavior of large amounts of liquids stored in containers under low-gravity conditions. Of particular concern is the motion of the liquids, and the resulting forces that are exerted on the container walls during a docking maneuver in orbit.

Until recently there have been no satisfactory analytical means for calculating the transient flow characteristics for liquids in containers. To develop such a means a research effort was undertaken to adapt the Marker-and-Cell (MAC) numerical technique (Ref. 1) to the axisymmetric and two-dimensional flow of liquids in containers. The capability for treating curved container boundaries was included in this adaptation. The resulting computer program will satisfactorily predict flow properties, including forces and moments on container walls. Results of this research effort are described in Ref. 2.

The success of the axisymmetric and two-dimensional liquid dynamics computations described in the preceding paragraph encouraged further effort to develop a three-dimensional liquid dynamics numerical technique. The research effort described in Ref. 2 was therefore extended to include this additional effort. Results of the extended research effort are described in this document.

Section 2
FORMULATION

2.1 GOVERNING EQUATIONS

The differential equations which govern the transient flow of a viscous incompressible fluid are

$$\frac{\partial \overline{v}}{\partial t} = -(\overline{v} \cdot \nabla)\,\overline{v} - \nabla\varphi + \nu\,\nabla^2\,\overline{v} + \overline{g} \tag{1}$$

$$D \equiv \nabla \cdot \overline{v} = 0 \tag{2}$$

where

$\overline{v}$ = velocity vector

φ = $\frac{p}{\rho}$ = pressure function

p = pressure

ρ = mass density

ν = kinematic viscosity coefficient

$\overline{g}$ = equivalent gravitational acceleration vector.

In Cartesian coordinates, Eqs. (1)and (2) may be written as

$$\left.\begin{aligned}
\frac{\partial u}{\partial t} &= -\frac{\partial u^2}{\partial x} - \frac{\partial uv}{\partial y} - \frac{\partial uw}{\partial z} - \frac{\partial \varphi}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + g_x \\
\frac{\partial v}{\partial t} &= -\frac{\partial uv}{\partial x} - \frac{\partial v^2}{\partial y} - \frac{\partial vw}{\partial z} - \frac{\partial \varphi}{\partial y} + \nu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial z^2}\right) + g_y \\
\frac{\partial w}{\partial t} &= -\frac{\partial uw}{\partial x} - \frac{\partial vw}{\partial y} - \frac{\partial w^2}{\partial z} - \frac{\partial \varphi}{\partial z} + \nu\left(\frac{\partial^2 w}{\partial x^2} \quad \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2}\right) + g_z
\end{aligned}\right\} \tag{3}$$

and

$$D \equiv \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \tag{4}$$

where u, v, w and g_x, g_y, g_z are the components of vectors $\bar{v}$ and $\bar{g}$ in the x, y, z-directions, respectively. The stress tensor of the system is

$$\left.\begin{aligned}
\sigma_{xx} &= -\varphi + 2\nu \frac{\partial u}{\partial x} \\
\sigma_{yy} &= -\varphi + 2\nu \frac{\partial v}{\partial y} \\
\sigma_{zz} &= -\varphi + 2\nu \frac{\partial w}{\partial z} \\
\sigma_{xy} &= \nu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \\
\sigma_{yz} &= \nu \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right) \\
\sigma_{zx} &= \nu \left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right)
\end{aligned}\right\} \tag{5}$$

In this study, the fluid is considered to be at rest initially; i.e., $\bar{v}(x, y, z, o) = p(x, y, z, o) = 0$. The boundary conditions of the fluid are

$$v_n = \sigma_t = 0 \qquad \text{(at a rigid boundary)} \tag{6}$$

$$\sigma_n = \sigma_t = 0 \qquad \text{(at a free surface)} \tag{7}$$

where a variable with a subscript n or t denotes the normal or tangential component of the variable, respectively.

2.2 FINITE DIFFERENCE FORMULATION

Cubic meshes are employed in writing the finite difference equations of the formulated problem. The velocity components and the pressure of a fluid are specified at the boundaries and at the center of a cell, respectively. As shown in Fig. 1, Eqs. (3) and (4) of cell (i, j, k) may be expressed in the following form:

$$\frac{1}{\delta t}\left(u^{n+1}_{i+\frac{1}{2},j,k} - u_{i+\frac{1}{2},j,k}\right)$$

$$= -\frac{1}{\delta x}\left[\left(u_{i+1,j,k}\right)^2 - \left(u_{i,j,k}\right)^2\right] - \frac{1}{\delta y}\left[(uv)_{i+\frac{1}{2},j+\frac{1}{2},k} - (u\ v)_{i+\frac{1}{2},j-\frac{1}{2},k}\right]$$

$$- \frac{1}{\delta z}\left[(uw)_{i+\frac{1}{2},j,k+\frac{1}{2}} - (uw)_{i+\frac{1}{2},j,k-\frac{1}{2}}\right] - \frac{1}{\delta x}\left(\varphi_{i+1,j,k} - \varphi_{i,j,k}\right)$$

$$+ \nu\left[\frac{1}{\delta x^2}\left(u_{i+\frac{1}{2},j,k} + u_{i-\frac{1}{2},j,k} - 2u_{i+\frac{1}{2},j,k}\right)\right.$$

$$+ \frac{1}{\delta y^2}\left(u_{i+\frac{1}{2},j+1,k} + u_{i+\frac{1}{2},j-1,k} - 2u_{i+\frac{1}{2},j,k}\right)$$

$$\left. + \frac{1}{\delta z^2}\left(u_{i+\frac{1}{2},j,k+1} + u_{i+\frac{1}{2},j,k-1} - 2u_{i+\frac{1}{2},j,k}\right)\right] + g_x \tag{8a}$$

$$\frac{1}{\delta t}\left(v^{n+1}_{i,j+\frac{1}{2},k} - v_{i,j+\frac{1}{2},k}\right)$$

$$= -\frac{1}{\delta x}\left[(uv)_{i+\frac{1}{2},j+\frac{1}{2},k} - (uv)_{i-\frac{1}{2},j+\frac{1}{2},k}\right] - \frac{1}{\delta y}\left[\left(v_{i,j+1,k}\right)^2 - \left(v_{i,j,k}\right)^2\right]$$

$$- \frac{1}{\delta z}\left[(vw)_{i,j+\frac{1}{2},k+\frac{1}{2}} - (vw)_{i,j+\frac{1}{2},k-\frac{1}{2}}\right] - \frac{1}{\delta y}\left(\varphi_{i,j+1,k} - \varphi_{i,j,k}\right)$$

$$+ \nu \left[\frac{1}{\delta x^2} (v_{i+1, j+\frac{1}{2}, k} + v_{i-1, j+\frac{1}{2}, k} - 2 v_{i, j+\frac{1}{2}, k}) \right.$$

$$+ \frac{1}{\delta y^2} (v_{i, j+\frac{3}{2}, k} + v_{i, j-\frac{1}{2}, k} - 2 v_{i, j+\frac{1}{2}, k})$$

$$\left. + \frac{1}{\delta z^2} (v_{i, j+\frac{1}{2}, k+1} + v_{i, j+\frac{1}{2}, k-1} - 2 v_{i, j+\frac{1}{2}, k}) \right] + g_y \tag{8b}$$

$$\frac{1}{\delta t} (w^{n+1}_{i, j, k+\frac{1}{2}} - w_{i, j, k+\frac{1}{2}})$$

$$= - \frac{1}{\delta x} \left[(uw)_{i+\frac{1}{2}, j, k+\frac{1}{2}} - (uw)_{i-\frac{1}{2}, j, k+\frac{1}{2}} \right] - \frac{1}{\delta y} \left[(vw)_{i, j+\frac{1}{2}, k+\frac{1}{2}} \right.$$

$$\left. - (vw)_{i, j-\frac{1}{2}, k+\frac{1}{2}} \right] - \frac{1}{\delta z} \left[(w_{i, j, k+1})^2 - (w_{i, j, k})^2 \right] - \frac{1}{\delta z} (\varphi_{i, j, k+1}$$

$$- \varphi_{i, j, k}) + \nu \left[\frac{1}{\delta x^2} (w_{i+1, j, k+\frac{1}{2}} + w_{i-1, j, k+\frac{1}{2}} - 2 w_{i, j, k+\frac{1}{2}}) \right.$$

$$+ \frac{1}{\delta y^2} (w_{i, j+1, k+\frac{1}{2}} + w_{i, j-1, k+\frac{1}{2}} - 2 w_{i, j, k+\frac{1}{2}})$$

$$\left. + \frac{1}{\delta z^2} (w_{i, j, k+\frac{3}{2}} + w_{i, j, k-\frac{1}{2}} - 2 w_{i, j, k+\frac{1}{2}}) \right] + g_z \tag{8c}$$

$$D_{i, j, k} = \frac{1}{\delta x} (u_{i+\frac{1}{2}, j, k} - u_{i-\frac{1}{2}, j, k}) + \frac{1}{\delta y} (v_{i, j+\frac{1}{2}, k} - v_{i, j-\frac{1}{2}, k})$$

$$+ \frac{1}{\delta z} (w_{i, j, k+\frac{1}{2}} - w_{i, j, k-\frac{1}{2}}) \tag{9}$$

where

$$(u_{i,j,k})^2 = u_{i+\frac{1}{2},j,k}\, u_{i-\frac{1}{2},j,k}$$

$$(v_{i,j,k})^2 = v_{i,j+\frac{1}{2},k}\, v_{i,j-\frac{1}{2},k}$$

$$(w_{i,j,k})^2 = w_{i,j,k+\frac{1}{2}}\, w_{i,j,k-\frac{1}{2}}$$

$$(uv)_{i+\frac{1}{2},j+\frac{1}{2},k} = \frac{1}{4}\,(u_{i+\frac{1}{2},j+1,k} + u_{i+\frac{1}{2},j,k})\,(v_{i+1,j+\frac{1}{2},k} + v_{i,j+\frac{1}{2},k})$$

$$(vw)_{i,j+\frac{1}{2},k+\frac{1}{2}} = \frac{1}{4}\,(v_{i,j+\frac{1}{2},k+1} + v_{i,j+\frac{1}{2},k})\,(w_{i,j+1,k+\frac{1}{2}} + w_{i,j,k+\frac{1}{2}})$$

$$(wu)_{i+\frac{1}{2},j,k+\frac{1}{2}} = \frac{1}{4}\,(w_{i+1,j,k+\frac{1}{2}} + w_{i,j,k+\frac{1}{2}})\,(u_{i+\frac{1}{2},j,k+1} + u_{i+\frac{1}{2},j,k}).$$

Note that the superscript $n+1$ refers to values at time $(n+1)\,\delta t$. Where there is no superscript it is understood that reference is made to values at time $n\,\delta t$.

The boundary conditions at a free surface can be written as

$$\varphi_{i,j,k} = \frac{2\nu}{\delta x}\,(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}) \qquad (\sigma_{xx} = 0) \qquad \text{(10a)}$$

$$\varphi_{i,j,k} = \frac{2\nu}{\delta y}\,(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}) \qquad (\sigma_{yy} = 0) \qquad \text{(10b)}$$

$$\varphi_{i,j,k} = \frac{2\nu}{\delta z}\,(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}) \qquad (\sigma_{zz} = 0) \qquad \text{(10c)}$$

$$\frac{1}{\delta y}\left(u_{i+\frac{1}{2},j+1,k} - u_{i+\frac{1}{2},j,k}\right) + \frac{1}{\delta x}\left(v_{i+1,j+\frac{1}{2},k} - v_{i,j+\frac{1}{2},k}\right) = 0$$

$$(\sigma_{xy} = 0) \qquad (10d)$$

$$\frac{1}{\delta z}\left(v_{i,j+\frac{1}{2},k+1} - v_{i,j+\frac{1}{2},k}\right) + \frac{1}{\delta y}\left(w_{i,j+1,k+\frac{1}{2}} - w_{i,j,k+\frac{1}{2}}\right) = 0$$

$$(\sigma_{yz} = 0) \qquad (10e)$$

$$\frac{1}{\delta x}\left(w_{i+1,j,k+\frac{1}{2}} - w_{i,j,k+\frac{1}{2}}\right) + \frac{1}{\delta z}\left(u_{i+\frac{1}{2},j,k+1} - u_{i+\frac{1}{2},j,k}\right) = 0$$

$$(\sigma_{zx} = 0) \qquad (10f)$$

Section 3
NUMERICAL SCHEMES

3.1 THE MAC COMPUTING TECHNIQUE

Similar to Eqs. (8a), (8b) and (8c), the u, v and w velocity components of cells (i-1, j, k), (i, j-1, k) and (i, j, k-1), respectively, can be found. Substituting these two sets of equations into Eq. (9) and assuming $D^{n+1}_{i,j,k} = 0$, a tentative pressure field (in terms of φ) of the fluid at time $(n+1)\delta$[illegible]s o[illegible]

$$\frac{1}{\delta x^2}(\varphi_{i+1,j,k} + \varphi_{i-1,j,k} - 2\varphi_{i,j,k}) + \frac{1}{\delta y^2}(\varphi_{i,j+1,k} + \varphi_{i,j-1,k} - 2\varphi_{i,j,k})$$

$$+ \frac{1}{\delta z^2}(\varphi_{i,j,k+1} + \varphi_{i,j,k-1} - 2\varphi_{i,j,k}) = \frac{D_{i,j,k}}{\delta t} - Q_{i,j,k} + \tilde{D}_{i,j,k} \tag{11}$$

where

$$Q_{i,j,k} = \frac{1}{\delta x^2}\left[(u_{i+1,j,k})^2 + (u_{i-1,j,k})^2 - 2(u_{i,j,k})^2\right]$$

$$+ \frac{1}{\delta y^2}\left[(v_{i,j+1,k})^2 + (v_{i,j-1,k})^2 - 2(v_{i,j,k})^2\right]$$

$$+ \frac{1}{\delta z^2}\left[(w_{i,j,k+1})^2 + (w_{i,j,k-1})^2 - 2(w_{i,j,k})^2\right]$$

$$+ \frac{2}{\delta x\,\delta y}\left[(uv)_{i+\frac{1}{2},j+\frac{1}{2},k} + (uv)_{i-\frac{1}{2},j-\frac{1}{2},k} - (uv)_{i+\frac{1}{2},j-\frac{1}{2},k}\right.$$

$$\left. - (uv)_{i-\frac{1}{2},j+\frac{1}{2},k}\right]$$

$$+ \frac{2}{\delta x \delta z} \left[(uw)_{i+\frac{1}{2}, j, k+\frac{1}{2}} + (uw)_{i-\frac{1}{2}, j, k-\frac{1}{2}} - (uw)_{i+\frac{1}{2}, j, k-\frac{1}{2}} \right.$$

$$\left. - (uw)_{i-\frac{1}{2}, j, k+\frac{1}{2}} \right]$$

$$+ \frac{2}{\delta y \delta z} \left[(vw)_{i, j+\frac{1}{2}, k+\frac{1}{2}} + (vw)_{i, j-\frac{1}{2}, k-\frac{1}{2}} - (vw)_{i, j+\frac{1}{2}, k-\frac{1}{2}} \right.$$

$$\left. - (vw)_{i, j-\frac{1}{2}, k+\frac{1}{2}} \right] \tag{12}$$

and

$$\widetilde{D}_{i,j,k} = \nu \left[\frac{1}{\delta x^2} (D_{i+1,j,k} + D_{i-1,j,k} - 2 D_{i,j,k}) + \frac{1}{\delta y^2} (D_{i,j+1,k} \right.$$

$$+ D_{i,j-1,k} - 2 D_{i,j,k}) + \frac{1}{\delta z^2} (D_{i,j,k+1} + D_{i,j,k-1}$$

$$\left. - 2 D_{i,j,k}) \right] \tag{13}$$

To find the correct pressure field of the fluid at time $(n+1)\,\delta t$, an iterative process is used. Using Eq. (11) the pressure of cell (i, j, k) at the completion of $(h+1)^{th}$ iteration is expressed as

$$\varphi^{h+1}_{i,j,k} = \frac{1+\alpha}{2\left(\frac{1}{\delta x^2} + \frac{1}{\delta y^2} + \frac{1}{\delta z^2}\right)} \left[\frac{1}{\delta x^2} (\varphi^{h}_{i+1,j,k} + \varphi^{h+1}_{i-1,j,k}) \right.$$

$$+ \frac{1}{\delta y^2} (\varphi^{h}_{i,j+1,k} + \varphi^{h+1}_{i,j-1,k}) + \frac{1}{\delta z^2} (\varphi^{h}_{i,j,k+1} + \varphi^{h+1}_{i,j,k-1})$$

$$\left. - \widetilde{Q}_{i,j,k} \right] - \alpha \varphi^{h}_{i,j,k} \tag{14}$$

where α is an over-relaxation parameter and

$$\tilde{Q}_{i,j,k} = \frac{D_{i,j,k}}{\delta t} - Q_{i,j,k} + \tilde{D}_{i,j,k} \tag{15}$$

The value of α can be taken between 0 and 1, and it is introduced for speeding up the iteration process. Hence, the MAC method calculates a set of velocities at time $(n+1)\,\delta t$ from the ~~velocities~~ and pressure at time $n\delta t$. Then, using the incompressibility property and the boundary conditions, an iterative process is used to compute the pressure of the fluid at time $(n+1)\,\delta t$. The procedure is repeated in the next computing cycle.

The major steps of a MAC computing cycle are:

<u>Step 1:</u> Use Eqs. (8a), (8b) and (8c) to calculate a set of velocities $u^{n+1}_{i+\frac{1}{2},j,k}$, $v^{n+1}_{i,j+\frac{1}{2},k}$ and $w^{n+1}_{i,j,k+\frac{1}{2}}$, then compute $\tilde{Q}_{i,j,k}$ from Eqs. (9), (12), (13) and (15).

<u>Step 2:</u> Iterate the pressure field, $\varphi^{h+1}_{i,j,k}$, until the following equation is satisfied throughout the entire flow field

$$\frac{\left|\,\left|\varphi^{h+1}_{i,j,k}\right| - \left|\varphi^{h}_{i,j,k}\right|\,\right|}{\left|\varphi^{h+1}_{k,j,k}\right| + \left|\varphi^{h}_{i,j,k}\right|} \le \epsilon_{\varphi}$$

where ϵ_{φ} is a small constant which is chosen to provide the necessary accuracy of a solution.

<u>Step 3:</u> Displace the marker particles according to their local velocities.

<u>Step 4:</u> Adjust the boundary velocities and pressure of the newly obtained flow field.

3.2 BOUNDARY CONDITIONS

To facilitate the construction of the numerical scheme and direction of the execution sequence, the following four types of cells are defined for identifying the status of a cell in a computing cycle:

1. Empty cell (E): A cell having no marker particle,
2. Boundary cell (B): An empty cell whose boundary face forms a portion of the wall of a container,
3. Surface cell (S): A cell having marker particles and neighboring with at least one empty cell, and
4. Full cell (F): A cell having marker particles and not neighboring with any empty cell.

Since there are many possible arrangements of empty cells around a surface cell, a surface cell is further classified to 63 cases (see Fig. 2).

The velocities of a boundary cell are calculated to satisfy the conditions of a rigid smooth wall. The velocities of a surface cell are computed to satisfy the incompressibility property and the free surface condition of the fluid. Equations for computing the velocities of a surface cell are given below, and the appropriate equations to be used in each case are listed in Table 1.

$$u_{i+\frac{1}{2},j,k} = u_{i-\frac{1}{2},j,k} - \frac{\delta x}{\delta y}(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}) - \frac{\delta x}{\delta z}(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}) \quad (16)$$

$$w_{i,j,k+\frac{1}{2}} = w_{i,j,k-\frac{1}{2}} - \frac{\delta z}{\delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}) - \frac{\delta z}{\delta y}(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}) \quad (17)$$

$$v_{i,j+\frac{1}{2},k} = v_{i,j-\frac{1}{2},k} - \frac{\delta y}{\delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}) - \frac{\delta y}{\delta z}(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}) \quad (18)$$

$$u_{i-\frac{1}{2},j,k} = u_{i+\frac{1}{2},j,k} + \frac{\delta x}{\delta y}(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}) + \frac{\delta x}{\delta z}(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}) \quad (19)$$

$$w_{i,j,k-\frac{1}{2}} = w_{i,j,k+\frac{1}{2}} + \frac{\delta z}{\delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}) + \frac{\delta z}{\delta y}(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}) \quad (20)$$

Table 1

EQUATIONS TO BE USED IN EACH CASE

Cases	Equations to be Used
1, 5, 11, 15, 49, 59	(16)
2, 7, 10, 50, 55, 58, 63	(17)
4, 14, 52, 62	(19)
8, 13, 56, 61	(20)
16, 21, 26, 31, 48, 53	(18)
32, 37, 42, 47	(21)
19	(22), (23) and (24)
22	(23), (24) and (25)
25	(22), (23) and (27)
28	(23), (25) and (27)
35	(22), (24) and 26)
38	(24), (25) and (26)
41	(22), (26) and (27)
44	(25), (26) and (27)
3, 51	(22) and (17)
6, 54	(25) and (17)
9, 57	(22) and (20)
12, 60	(25) and (20)
17, 27	(22) and (18)
18, 23	(23) and (17)
20, 30	(25) and (18)
24, 29	(23) and (20)
33, 43	(22) and (21)
34, 39	(26) and (17)
36, 46	(25) and (21)
40, 45	(26) and (20)

$$v_{i,j-\frac{1}{2},k} = v_{i,j+\frac{1}{2},k} + \frac{\delta y}{\delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}) + \frac{\delta y}{\delta z}(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}) \quad (21)$$

$$u_{i+\frac{1}{2},j,k} = u_{i-\frac{1}{2},j,k} \quad (22)$$

$$v_{i,j+\frac{1}{2},k} = v_{i,j-\frac{1}{2},k} \quad (23)$$

$$w_{i,j,k+\frac{1}{2}} = w_{i,j,k-\frac{1}{2}} \quad (24)$$

$$u_{i-\frac{1}{2},j,k} = u_{i+\frac{1}{2},j,k} \quad (25)$$

$$v_{i,j-\frac{1}{2},k} = v_{i,j+\frac{1}{2},k} \quad (26)$$

$$w_{i,j,k-\frac{1}{2}} = w_{i,j,k+\frac{1}{2}} \quad (27)$$

In a three-dimensional MAC formulation, there are also 12 possible configurations involving two empty cells neighboring with a free surface (Fig. 3). The velocity component between these two empty cells needs to be considered in order to preserve the no-shear stress condition at a free surface. The following equations are used for calculating these velocities:

Case	Equation to be Used
1	$u_{i+\frac{1}{2},j,k+1} = u_{i+\frac{1}{2},j,k} - \frac{\delta z}{\delta x}(w_{i+1,j,k+\frac{1}{2}} - w_{i,j,k+\frac{1}{2}})$
2	$u_{i+\frac{1}{2},j,k-1} = u_{i+\frac{1}{2},j,k} + \frac{\delta z}{\delta x}(w_{i+1,j,k-\frac{1}{2}} - w_{i,j,k-\frac{1}{2}})$
3	$u_{i+\frac{1}{2},j+1,k} = u_{i+\frac{1}{2},j,k} - \frac{\delta y}{\delta x}(v_{i+1,j+\frac{1}{2},k} - v_{i,j+\frac{1}{2},k})$
4	$u_{i+\frac{1}{2},j-1,k} = u_{i+\frac{1}{2},j,k} + \frac{\delta y}{\delta x}(v_{i+1,j-\frac{1}{2},k} - v_{i,j-\frac{1}{2},k})$

5 $$v_{i-1,j+\frac{1}{2},k} = v_{i,j+\frac{1}{2},k} + \frac{\delta x}{\delta y}(u_{i-\frac{1}{2},j+1,k} - u_{i-\frac{1}{2},j,k})$$

6 $$v_{i+1,j+\frac{1}{2},k} = v_{i,j+\frac{1}{2},k} - \frac{\delta x}{\delta y}(u_{i+\frac{1}{2},j+1,k} - u_{i+\frac{1}{2},j,k})$$

7 $$v_{i,j+\frac{1}{2},k+1} = v_{i,j+\frac{1}{2},k} - \frac{\delta z}{\delta y}(w_{i,j+1,k+\frac{1}{2}} - w_{i,j,k+\frac{1}{2}})$$

8 $$v_{i,j+\frac{1}{2},k-1} = v_{i,j+\frac{1}{2},k} + \frac{\delta z}{\delta y}(w_{i,j+1,k-\frac{1}{2}} - w_{i,j,k-\frac{1}{2}})$$

9 $$w_{i-1,j,k+\frac{1}{2}} = w_{i,j,k+\frac{1}{2}} + \frac{\delta x}{\delta z}(u_{i-\frac{1}{2},j,k+1} - u_{i-\frac{1}{2},j,k})$$

10 $$w_{i+1,j,k+\frac{1}{2}} = w_{i,j,k+\frac{1}{2}} - \frac{\delta x}{\delta z}(u_{i+\frac{1}{2},j,k+1} - u_{i+\frac{1}{2},j,k})$$

11 $$w_{i,j-1,k+\frac{1}{2}} = w_{i,j,k+\frac{1}{2}} + \frac{\delta y}{\delta z}(v_{i,j-\frac{1}{2},k+1} - v_{i,j-\frac{1}{2},k})$$

12 $$w_{i,j+1,k+\frac{1}{2}} = w_{i,j,k+\frac{1}{2}} - \frac{\delta y}{\delta z}(v_{i,j+\frac{1}{2},k+1} - v_{i,j+\frac{1}{2},k})$$

The pressure of a boundary cell is considered to be equal to the pressure of the neighboring full or surface cell. In general, a surface cell does not carry a pressure unless it neighbors with one and only one empty cell. As shown in Fig. 2, there are six cases of a surface cell which can have a pressure. These boundary pressures are defined below:

$$\varphi_{i,j,k} = \frac{2\nu}{\delta x}(u_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}) \quad \text{(for cases 1 and 4)}$$

$$\varphi_{i,j,k} = \frac{2\nu}{\delta y}(v_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}) \quad \text{(for cases 16 and 32)}$$

$$\varphi_{i,j,k} = \frac{2\nu}{\delta z}(w_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}) \quad \text{(for cases 2 and 8)}$$

3.3 DISPLACING OF MARKER PARTICLES

A volume-velocity weighting scheme is used to calculate the velocity of a marker particle. Figure 4 shows the mechanics of the scheme, and Table 2 gives the formulas for calculating the velocity components of a marker particle in cell (K, J, I) or (i, j, k). At the completion of a computing cycle each marker particle will be displaced by a distance $\overline{v}\,\delta t$. The cell status and the boundary velocities and pressure are then readjusted in accordance with the new flow field.

3.4 FORCES AND MOMENTS

The dynamic loads exerted by a moving liquid on its container are computed from the following surface integrals

$$\overline{F} = \int_S p\,\hat{n}\,dS$$

and

$$\overline{M} = \int_S p\,\overline{r} \times \hat{n}\,dS$$

where $\overline{F}$ and $\overline{M}$ are the force and moment vectors, respectively. Notations $\overline{r}$ and $\hat{n}$ are the position vector and unit normal of a fluid particle on the surface, respectively. In the MAC formulation, the integrals are evaluated by summing up the quantities contributed by all cells neighboring with the container wall. The loads are calculated at the end of one or several computing cycles about the X, Y, Z-coordinate system shown in Fig. 9.

Table 2

FORMULAS FOR CALCULATING VELOCITY COMPONENTS OF MARKER PARTICLE m IN CELL (K, J, I)

$x \le x_c$		$x > x_c$		$y \le y_c$		$y > y_c$		$z \le z_c$		$z > z_c$		To be used for Computing
x_1	L	x_1	L	y_1	M	y_1	M	z_1	N	z_1	N	
x + 2.0 - I	I - 1	x + 2.0 - I	I - 1	y + 2.5 - J	J - 1	J - 0.5 - y	J + 1	z + 2.5 - K	K - 1	K - 0.5 - z	K + 1	U_m
x + 2.5 - I	I - 1	I - 0.5 - x	I + 1	y + 2.0 - J	J - 1	y + 2.0 - J	J - 1	z + 2.5 - K	K - 1	K - 0.5 - z	K + 1	V_m
x + 2.5 - I	I - 1	I - 0.5 - x	I + 1	y + 2.5 - J	J - 1	J - 0.5 - y	J + 1	z + 2.0 - K	K - 1	z + 2.0 - K	K - 1	W_m

$$U_m = x_1 y_1 z_1 U(K,J,I) + x_2 y_1 z_1 U(K,J,L) + x_1 y_1 z_2 U(N,J,I) + x_2 y_1 z_2 U(N,J,L)$$
$$+ x_1 y_2 z_1 U(K,M,I) + x_2 y_2 z_1 U(K,M,L) + x_1 y_2 z_2 U(N,M,I) + x_2 y_2 z_2 U(N,M,L)$$

$$V_m = x_1 y_1 z_1 V(K,J,I) + x_1 y_2 z_1 V(K,M,I) + x_2 y_1 z_1 V(K,J,L) + x_2 y_2 z_1 V(K,M,L)$$
$$+ x_1 y_1 z_2 V(N,J,I) + x_1 y_2 z_2 V(N,M,I) + x_2 y_1 z_2 V(N,J,L) + x_2 y_2 z_2 V(N,M,L)$$

$$W_m = x_1 y_1 z_1 W(K,J,I) + x_1 y_1 z_2 W(N,J,I) + x_1 y_2 z_1 W(K,M,I) + x_1 y_2 z_2 W(N,M,I)$$
$$+ x_2 y_1 z_1 W(K,J,L) + x_2 y_1 z_2 W(N,J,L) + x_2 y_2 z_1 W(K,M,L) + x_2 y_2 z_2 W(N,M,L)$$

NOTE: $x_2 = 1 - x_1$, $y_2 = 1 - y_1$, $z_2 = 1 - z_1$.

x_c, y_c, z_c are the coordinates of cell center.

Section 4
THE COMPUTER PROGRAMS

4.1 USER'S GUIDE OF THE LHMAC2 PROGRAM

A detailed discussion of the Lockheed-Huntsville two-dimensional and axisymmetric MAC computer program (LHMAC2) is presented in Ref. 2. The program has been modified to include the capability of simulating the transient flow of a liquid in a shallow container. Variables to be used in preparing the data deck of the LHMAC2 program are defined in Table 3, and the sequence and format of the data cards are shown below:

Data Set	Format	Variables
1	2I5, 3F8.3, 2I5, F8.3	IBAR, JBAR, DR, DZ, DT, IPHM, PC, ALP
2	12A6	NAME
3	4F4.1, 8F8.3	BCB, BCR, BCT, BCL, A, B, C, NU, EPS, GR, GZ, VSCALE
4	4F10.3, 4I10	T, TWPLT, TWPRT, TWFIN, LPR, NPRT3
5	10I5, 2F8.3	TYPE, L1, L2, L3, L4, L5, L6, L7, NXB, NYB, UL, UR
6	16I5	NSGMTS, NJC1, NJC2, LQUDHT, NPRT2, ISUR, ICYCLE, IPLOT, NGLVL
7	8F10.3	(RCOORD(I), I = 1, NSGMTS)
8	8F10.3	(ZCOORD(I), I = 1, NSGMTS)
9	8F10.3	DEPS, VEPS, DBETA, SIGNVN, STH, STR, STZ, DS, COFST, RHO, THCKNS
10	8F10.3	(GLVLTT(I), I = 1, NGLVL2)*
11	8F10.3	(GRT(I), I = 1, NGLVL1)*
12	8F10.3	(GZT(I), I = 1, NGLVL1)
13	16I5	(JHYB(I), I = 1, NJCELL)**

---

* NGLVL2 = NGLVL + 1, NGLVL1 = NGLVL + NGLVL

** NJCELL = NJCI + NJC2

Data Set	Format	Variables
14	16I5	(LHYB(I), I = 1, NJCELL)
15	16I5	(NHYB(I), I = 1, NJCELL)
16	2I5, 6F8.3	NX, NY, XC, YC, XD, YD, UO, VO

Table 3

Variables to Be Used in Preparing the Data Deck of the LHMAC2 Program

Variable(s)	Description
A, B, C	0.0, 0.0, 0.0
ALP	0.0
BCB, BCR, BCT, BCL	1.0, 1.0, 1.0, 1.0
COFST	surface tension coefficient
DBETA	iteration step
DEPS	0.25
DR, DZ	mesh size in the r and z-directions, respectively
DS	0.4
DT	time step
EPS	0.0002
GLVLTT(I)	initial and final points of time intervals of the equivalent gravitational acceleration
GR, GZ	0.0, 0.0
GRT(I), GZT(I)	equivalent gravitational acceleration in the r and z-directions, respectively
IBAR, JBAR	number of interior cells in the r and z-directions, respectively (see Fig. 5)
IPHM	0
ICYCLE	4
ISUR	0, flat free surface; 1, curved initial free surface; 2, surface tension effect included

Variable(s)	Description
IPLOT	1
JHYB(I), LHYB(I), NHYB(I)	used for defining HS cells (see Fig. 5)
LQUDHT	liquid height (see Fig. 5)
LPR	0, no velocity plot; 1, having velocity plots
L1, L2, ..., L7	0, 0, ..., 0
NAME	title of a problem
NGLVL	number of equivalent gravitational acceleration intervals
NJC1, NJC2	for defining HS cells (see Fig. 5)
NPRT2, NPRT3	0, 0
NU	kinematic viscosity coefficient
NXB, NYB	0, 0
NX, NY	number of marker particles per cell in the r and z-directions, respectively
NSGMTS	number of segments
PC	0, axisymmetric container; 1, channel-type container
RCOORD(I), ZCOORD(I)	for describing container geometry (see Fig. 5) Limitation: ZCOORD(I) cannot be specified to cross a cell in the z-direction
RHO	mass density
SIGNVN	-1.0
STH, STR, STZ	for defining a curved free surface (see Fig. 5)
T, TWFIN	initial and final time points of a problem, respectively
THCKNS	1.0
TWPLT, TWPRT	time increments for plots and printout, respectively
TYPE	0, container with flat ends; 2, container with ellipsoidal ends
UL, UR, UO, VO	0.0, 0.0, 0.0, 0.0
VEPS	0.04
VSCALE	scale factor of velocity plots

Variable(s)	Description
(XC, YC), (XD, YD)	coordinates in meters of the lower left and upper right corners of a container having flat ends, respectively

Note that multiple runs can be made, and execution is terminated by setting IBAR = -1

Example: The data deck of Sample Problem I is shown below (also see Fig. 6)

```
  18    32 0.1875  0.1875 0.015625    30      0 0.0
PROPELLANT DYNAMICS -- CASE I  AXISYMMETRIC, H = 0.75 M
1.0 1.0 1.0 1.0 0.0       0.0       0.0       1.0  -6 0.0002  0.0       11.4      2.0
0.0          0.0625     0.25        1.01                 1         0
   2
  10    5     0     4     0     0     4     1     3
0.0          2.0         6.0         9.0         12.0        15.0        16.75       18.0
18.0         0.0
0.0          0.0         0.375       1.0         2.0         3.0         4.0         5.0
32.0         32.0
0.25         0.04        1.0        -1.0         4.0         27.0        3.0         0.4
0.000038     800.0       1.0
0.0          0.09375     0.71        1.125
-3.8        -3.3         0.0         0.0         0.0         0.0
11.4         11.4        11.4        11.4        0.0         0.0
    1     2     3     4     5
    1    10    13    16    17
    9     3     3     2     2
    5     5
    0
   -1
```

4.2 USER'S GUIDE OF THE LHMAC3 PROGRAM

The LHMAC3 program needs approximately, but no more than, 64K core space and four drum areas for temporary storage. The program can be used to simulate the flow in a rectangular container requiring up to 5000 cells. The output of the program will provide the following information:

- Plots of 3-D flow and 2-D velocity fields of a transient flow,
- Pressure distribution on the container wall, and
- Dynamic loads induced by the moving liquid.

A brief block diagram which shows the organization of the LHMAC3 program is given in Chart 1.

The sequence and format of preparing the data deck of the LHMAC3 program are given below, and variables to be used are defined in Table 4.

Data Set	Format	Variables
1	16I5	ITYPE, IBAR, JBAR, KBAR
2	12A6	TITLE
3	16I5	LNTH1, LNTH2, ..., LNTH7
4	16I5	NMPPUX, NMPPUY, NMPPUZ
5	16I5	(IOPT(I), I = 1,16), (IPLT(I), I = 1,16), (IPRT(I), I = 1,16)
6	16I5	NGRT, LHT, NVPLT, (NSEGV(I), I = 1, 3), (JPLANE(I), I = 1, 3)
7	16F5.1	(XV(J,I), I = 1, NSEGV(J)) } Repeat J = NVPLT times
8	16F5.1	(ZV(J,I), I = 1, NSEGV(J)) }
9	8F10.4	(BDRY(I), I = 1, 6)
10	8F10.4	(GRT(I), I = 1, J1)*
11	8F10.4	(GRX(I), I = 1, J2)*
12	8F10.4	(GRY(I), I = 1, J2)
13	8F10.4	(GRZ(I), I = 1, J2)
14	8F10.4	DT, DBETA, DX, DY, DZ, EPSA, EPSD, EPSP, EPSV, RHO, RNU, VSCALE, WALL
15	8F10.4	TIN, TPLT, TPRT, TCOMP, TFIN, TCPU

*J1 = NGRT + 1, J2 = NGRT + NGRT

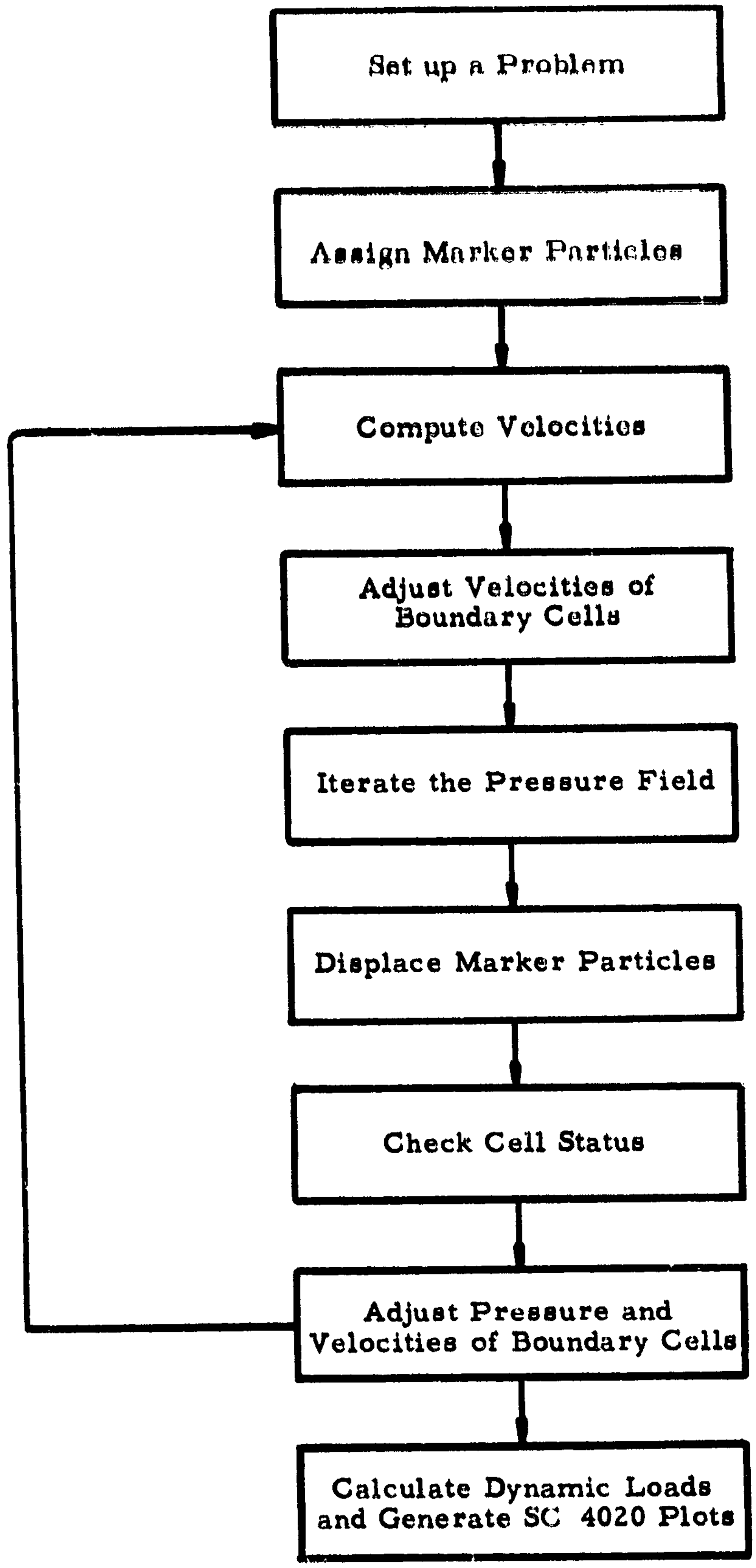


Chart 1 - Program Organization of the LHMAC3 Program

Table 4

Variables to be Used in Preparing the Data Deck of the LHMAC3 Program

Variable(s)	Description
BDRY(I)	1.0
DBETA	0.125
DT	time step
DX, DY, DZ	mesh size in the x, y and z-directions, respectively
EPSA, EPSD, EPSP, EPSV	0.5, 0.25, 0.0001, 0.04
GRT(I)	initial and final points of time intervals of the equivalent gravitational acceleration
GRX(I), GRY(I), GRZ(I)	equivalent gravitational acceleration in the x, y and z-directions, respectively
IBAR, JBAR, KBAR	number of interior cells in the x, y and z-directions, respectively
IOPT(I)	0 (or blank)
IPLT(I)	1 for I = 1, 2, blank for all others
IPRT(I)	1 for I = 1, 2, 3, 4, 5, blank for all others
ITYPE	1
JPLANE(I)	velocity projection on Y = Jth cell plane
LHT	liquid height*
LNTH1, LNTH2, LNTH3	equal to IBAR, JBAR and KBAR, respectively
LNTH4, ..., LNTH7	blank
NGRT	number of equivalent gravitational acceleration intervals
NMPPUX, NMPPUY, NMPPUZ	number of marker particles per cell in the x, y and z-directions, respectively
NSEGV(I)	4
NVPLT	number of velocity plots per time point (max. 3)
RHO	mass density
RNU	kinematic viscosity coefficient
TCOMP	time intervals for computing dynamic loads and pressure distribution on container wall

* Measured as number of cells

Variables	Description
TCPU	number of CPU time allowed for executing this run. It is used to save SC-4020 plots in case of max. time being reached.
TIN, TFIN	initial and final time points of a problem, respectively
TPLT, TPRT	time intervals for plots and print-out, respectively
TITLE	title of a problem
VSCALE	scale factor of velocity plots
WALL	1.0
XV(I, J), ZV(I, J)	coordinates of container geometry in the x and z-directions, respectively (measured as number of cells).

Note that multiple runs can be made, and execution is terminated by setting ITYPE = -1. Selection of time step DT is suggested to satisfy the following two conditions:

$$4 \left|\bar{v}\right| \delta t \leq \min(\delta x, \delta y, \delta z)$$

and

$$4\nu\delta t \leq \frac{(\delta x)^2(\delta y)^2 + (\delta y)^2(\delta z)^2 + (\delta z)^2(\delta x)^2}{(\delta x)^2 + (\delta y)^2 + (\delta z)^2}$$

Example: The data deck of Sample Problem II, Case 3, is shown below.

```
    1    16     8    14
PROPELLANT SLOSHING IN A RECTANGULAR TANK
   16     8    14
    2     2     2
    0
    1     1
    1     1     1     1     1
    2     8     1     4     4     4     5     7     8
0.0    16.0  16.0  0.0
0.0    0.0   14.0  14.0
1.0          1.0          1.0          1.0          1.0          1.0
0.0          0.5          5.0
5.0          1.0          1.0          1.0
0.0          0.0          0.0          0.0
0.0          0.0          0.0          0.0
0.0625       0.125        0.25         0.25         0.25         0.5          0.25         0.0001
0.04         1000.0       0.000001     0.25         1.0
0.0          0.0625       0.0625       0.0625       1.0          570.0
```

Section 5
EXAMPLES

<u>Sample Problem I</u>: Flow of a Viscous Incompressible Fluid in an Axisymmetric Container

The container geometry and liquid height of the problem is shown in Fig. 7. Material properties, time increment, iteration step and other parameters are given in the example of Section 4.1. Using the LHMAC2 program the first second of the flow under the following g-level is simulated:

$$g_r = 0$$

$$g_z = 11.4 \text{ m/sec}^2 \quad (0<t \leq 0.7 \text{ sec})$$

$$= 0 \quad (t>0.7 \text{ sec})$$

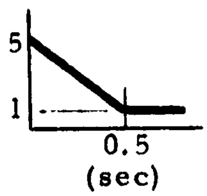
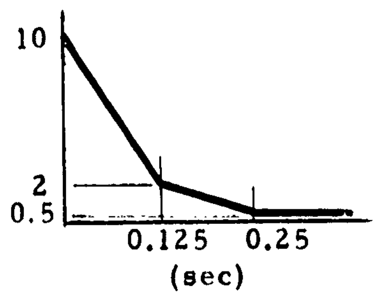
Fig. 8 shows the flow and velocity fields of the fluid at selected times.

<u>Sample Problem II:</u> Flow of a Viscous Incompressible Fluid in a Rectangular Container under an Arbitrary g-level

The flow of a viscous incompressible fluid in a rectangular container is used as a sample problem of the LHMAC3 program. Many cases are studied to show the influence of viscosity and g-level on a flow. Figure 9 shows the geometry of the problem. Parameters which are used in modeling this problem are given in the example of Section 4.2. Some of the cases which have been investigated are listed in Table 5. The flow and velocity fields of the fluid are shown in Fig. 10.

Table 5

VISCOSITY AND g-LEVEL USED IN VARIOUS CASES OF SAMPLE PROBLEM II

Case	Equivalent Gravitational Acceleration (m/sec^2)			Kinematic Viscosity Coefficient (m^2/sec)
	g_x	g_y	g_z	
1	5	0	-10	1×10^{-6}
2	5	0	10	1×10^{-6}
3		0	0	1×10^{-6}
4	5	5	10	1×10^{-6}
5	5	5	10	1×10^{-1}
6	5	5	10	1×10^{-3}
7			-0.1	1×10^{-6}

Section 6
CONCLUSIONS AND RECOMMENDATIONS

This research effort resulted in the development of analytical tools for the study of propellant motion in tanks during a docking maneuver in space. The axisymmetric flow computer program is capable of accurate simulation of propellant dynamics in realistic shaped containers for docking loads aligned parallel to the tank axis. For off-axis loads the two-dimensional program provides a reasonable indication of propellant motion and the forces and moments on the tank wall.

The pilot 3-D program can be used for the study of propellant dynamics in a rectangular shaped container and can be developed further to study problems of a more general nature.

Propellant dynamics problems to be encountered in the future space shuttle flights will be mostly of a 3-D nature. The lack of symmetry is due either to the tank geometry or the acceleration vector. Development of a computer program for investigating these problems is needed. Recommendations of continued research effort to extend the LHMAC3 program for studying the dynamics of liquid propellant in commonly used tanks are to:

- Include the capabilities of handling curved boundaries and arbitrary initial free surface into the LHMAC3 program.
- Refine the current 3-D MAC computing technique for analyzing flows having waved free surface.
- Develop a new numerical scheme of including the surface tension effect for flows in a low-g field.
- Study the roles of viscosity and tank geometry in flows under a wide range of g-levels.

REFERENCES

1. Welch, J. E., F. H. Harlow, J. P. Shannon, and B. J. Daly, "The MAC Method, A Computing Technique for Solving Viscous, Incompressible, Transient Fluid-Flow Problems Involving Free Surface," LA-3425, Los Alamos Scientific Laboratory, University of California, Los Alamos, N. M., January 1969.

2. Feng, G. C., and S. J. Robertson, "Study on Propellant Dynamics During Docking – Interim Report," LMSC-HREC D225157, Lockheed Missiles & Space Company, Huntsville, Ala., June 1971.

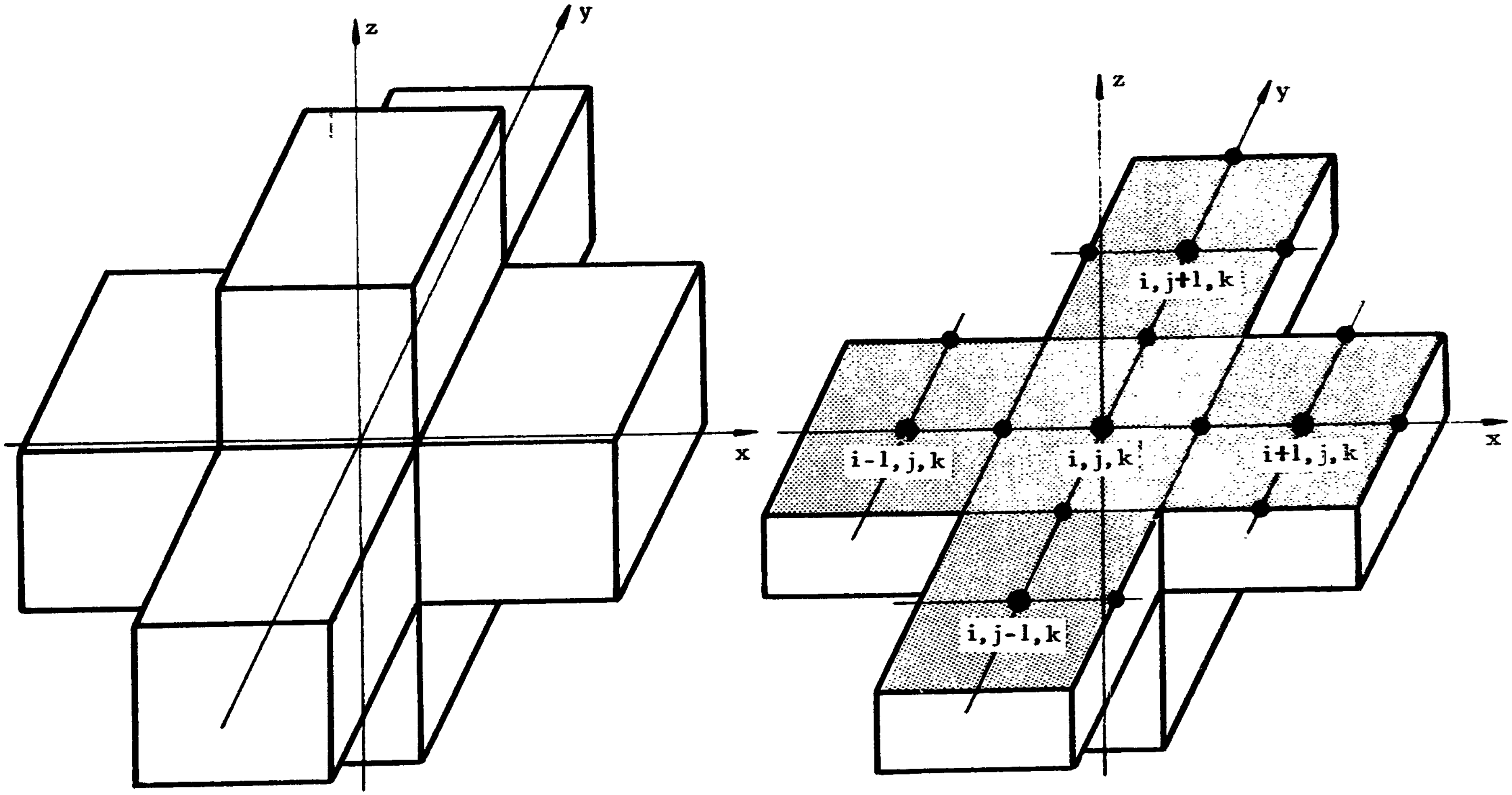


Fig. 1 - Cubic Meshes to be Used in Specifying the Pressure and the Velocity Fields of a Liquid

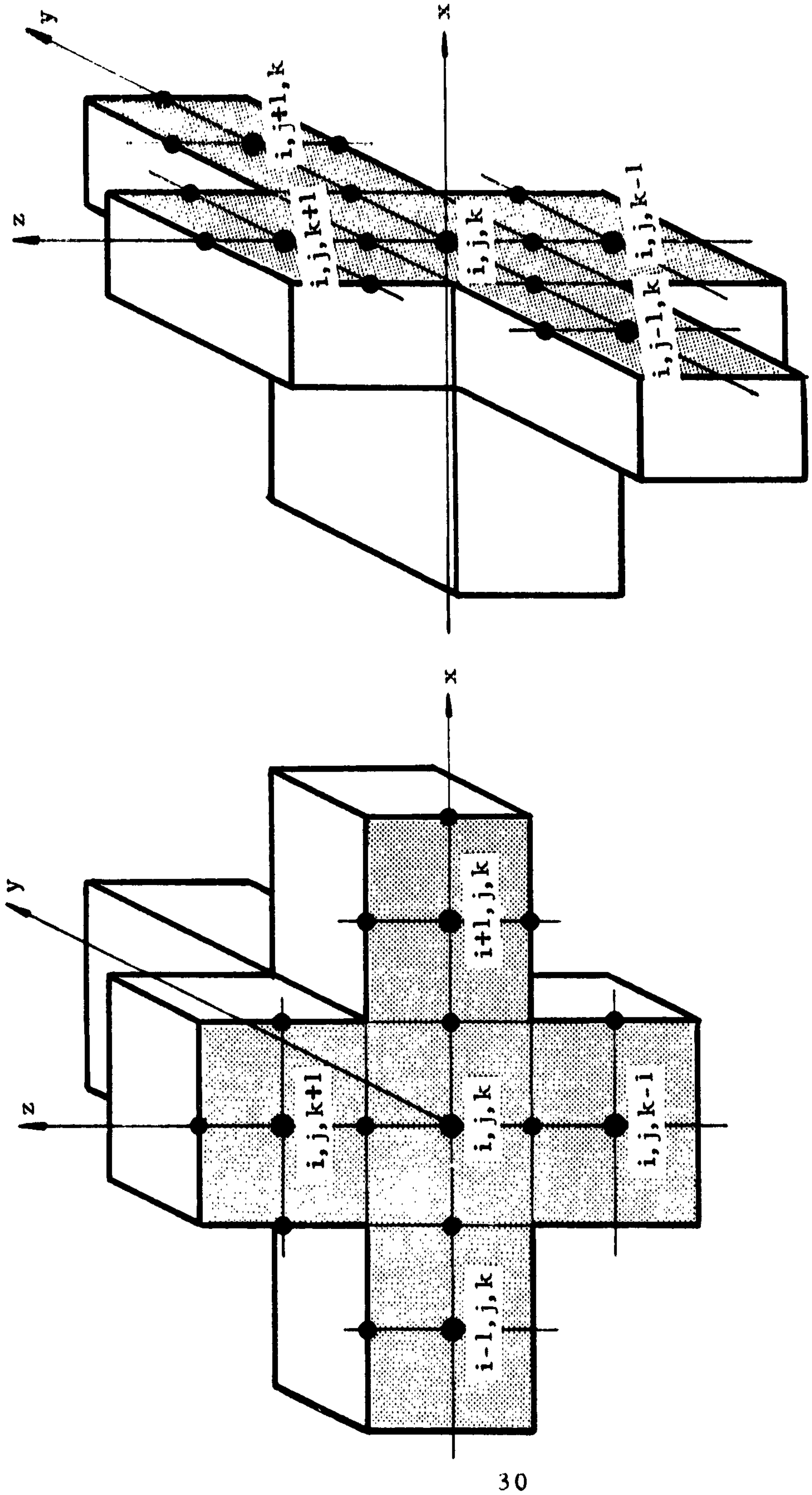


Fig. 1 - (Concluded)

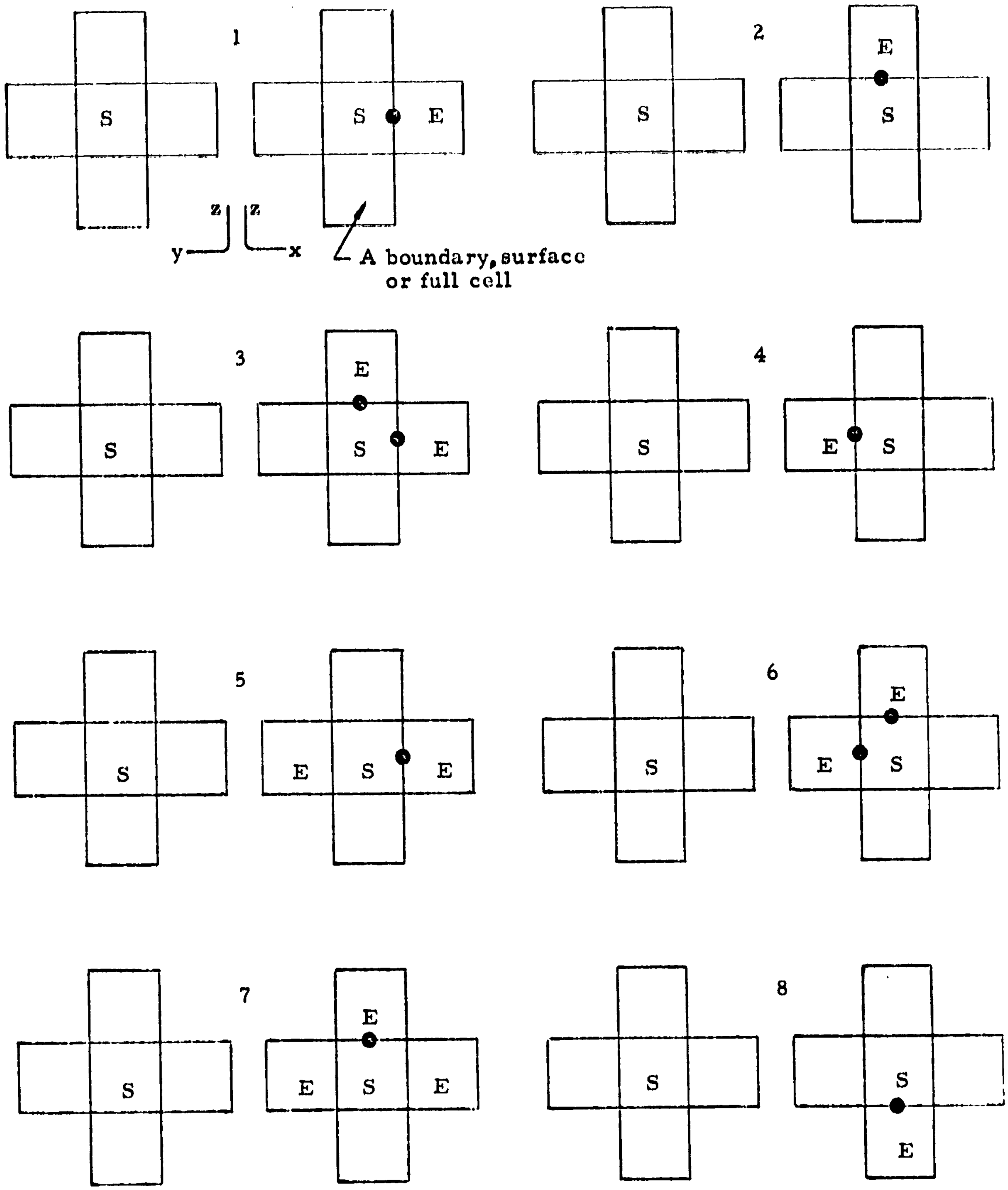


Fig. 2 - Possible Arrangements of Empty Cells Around a Surface Cell

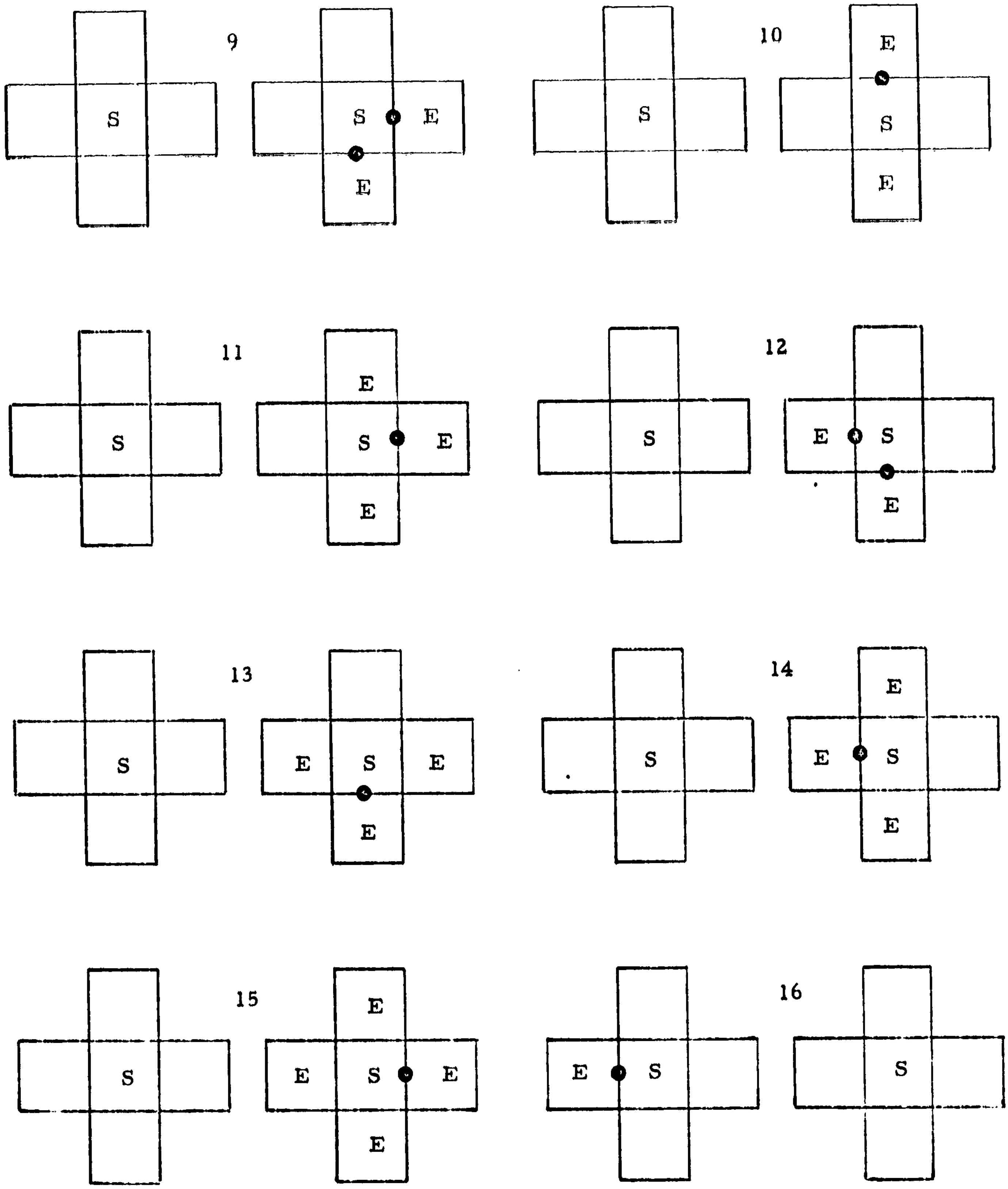


Fig. 2 - (Continued)

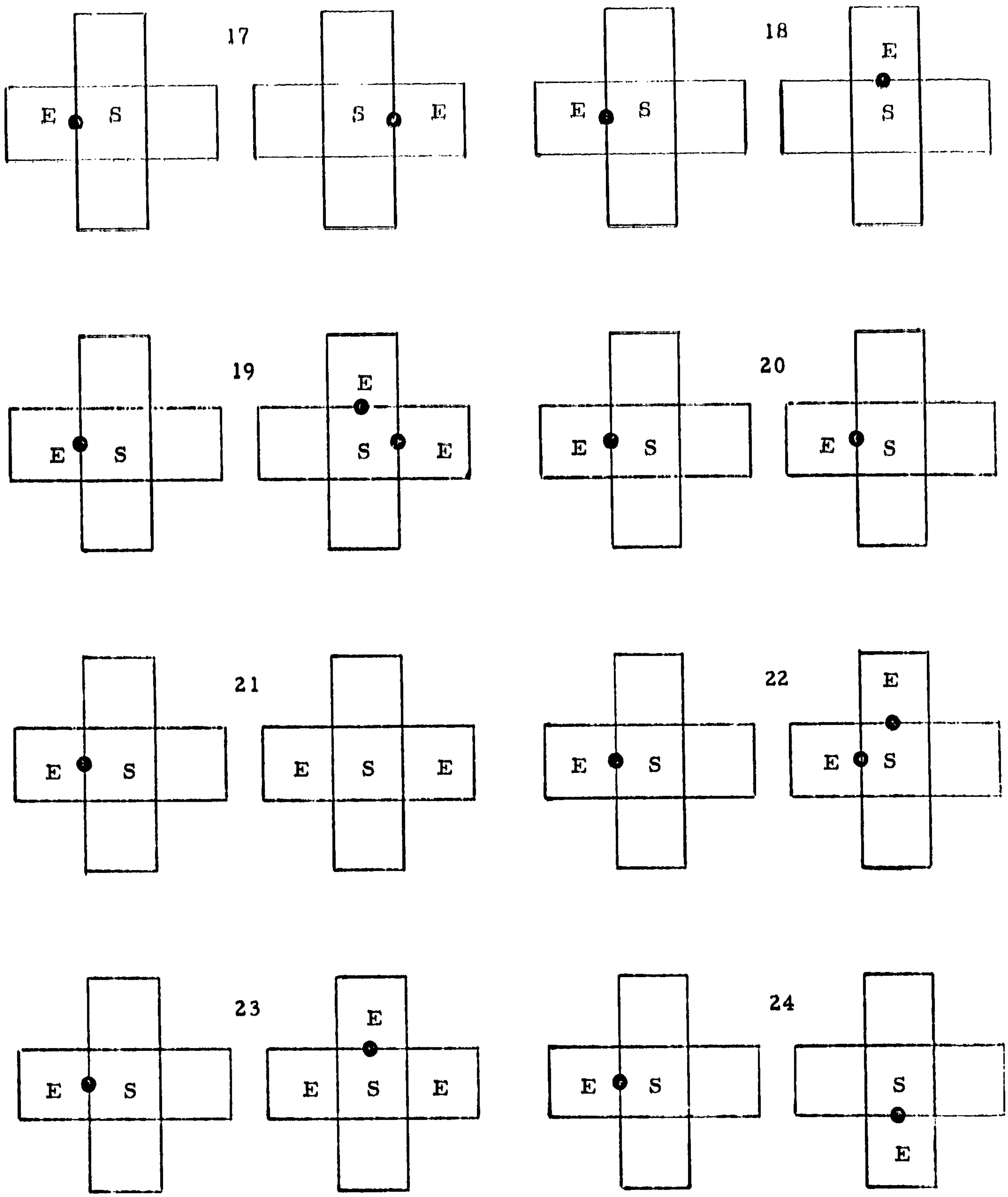


Fig. 2 - (Continued)

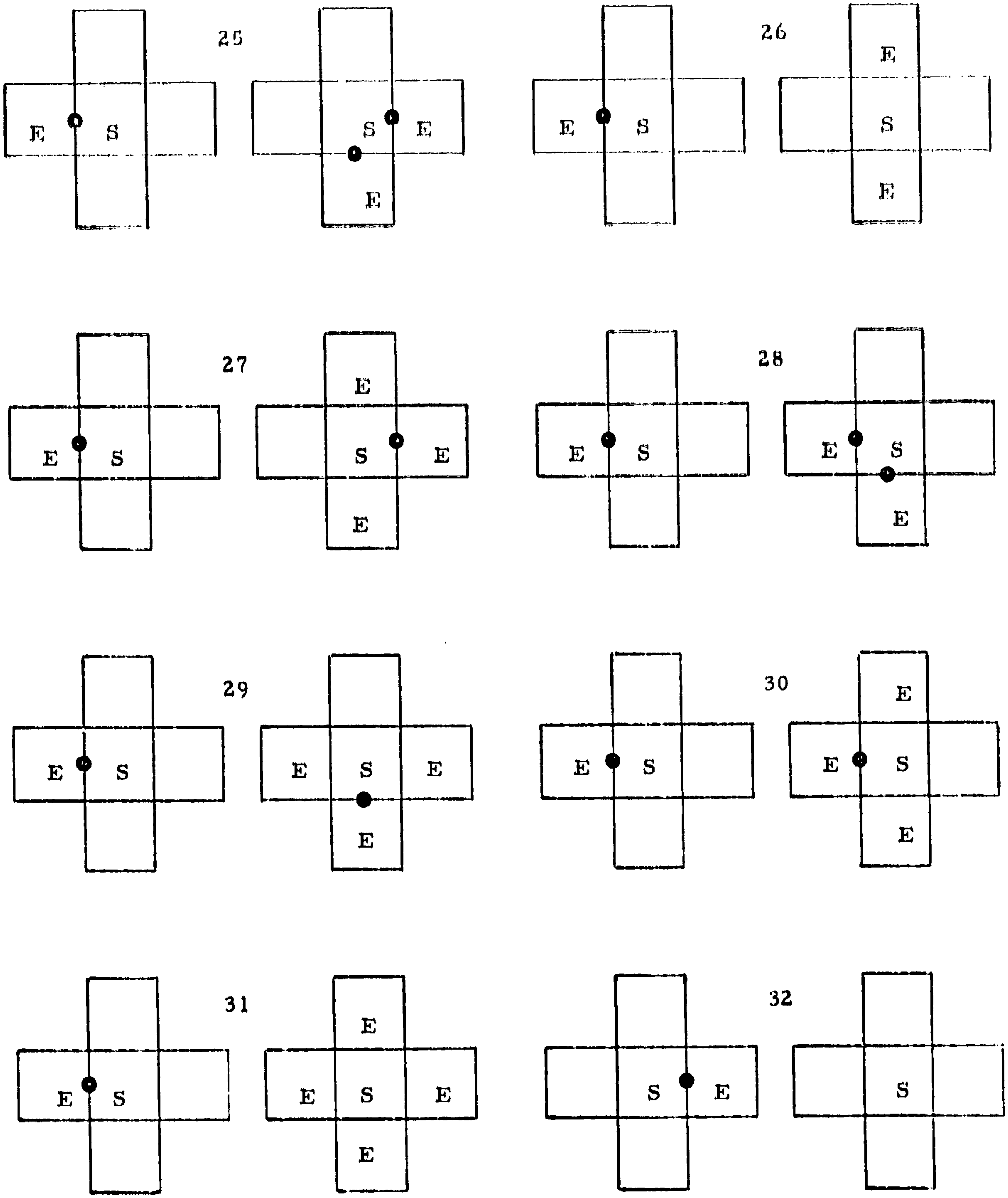


Fig. 2 - (Continued)

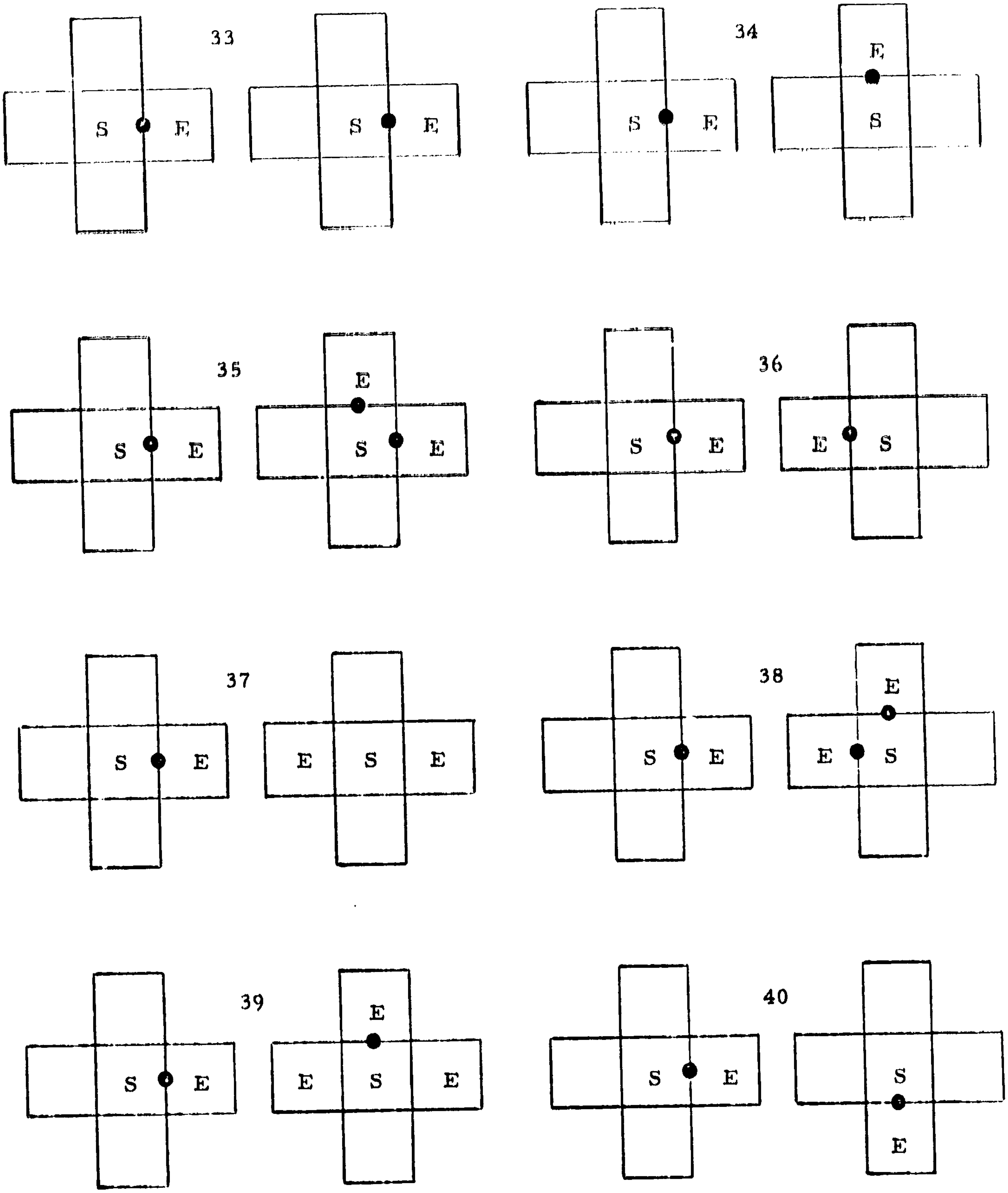


Fig. 2 - (Continued)

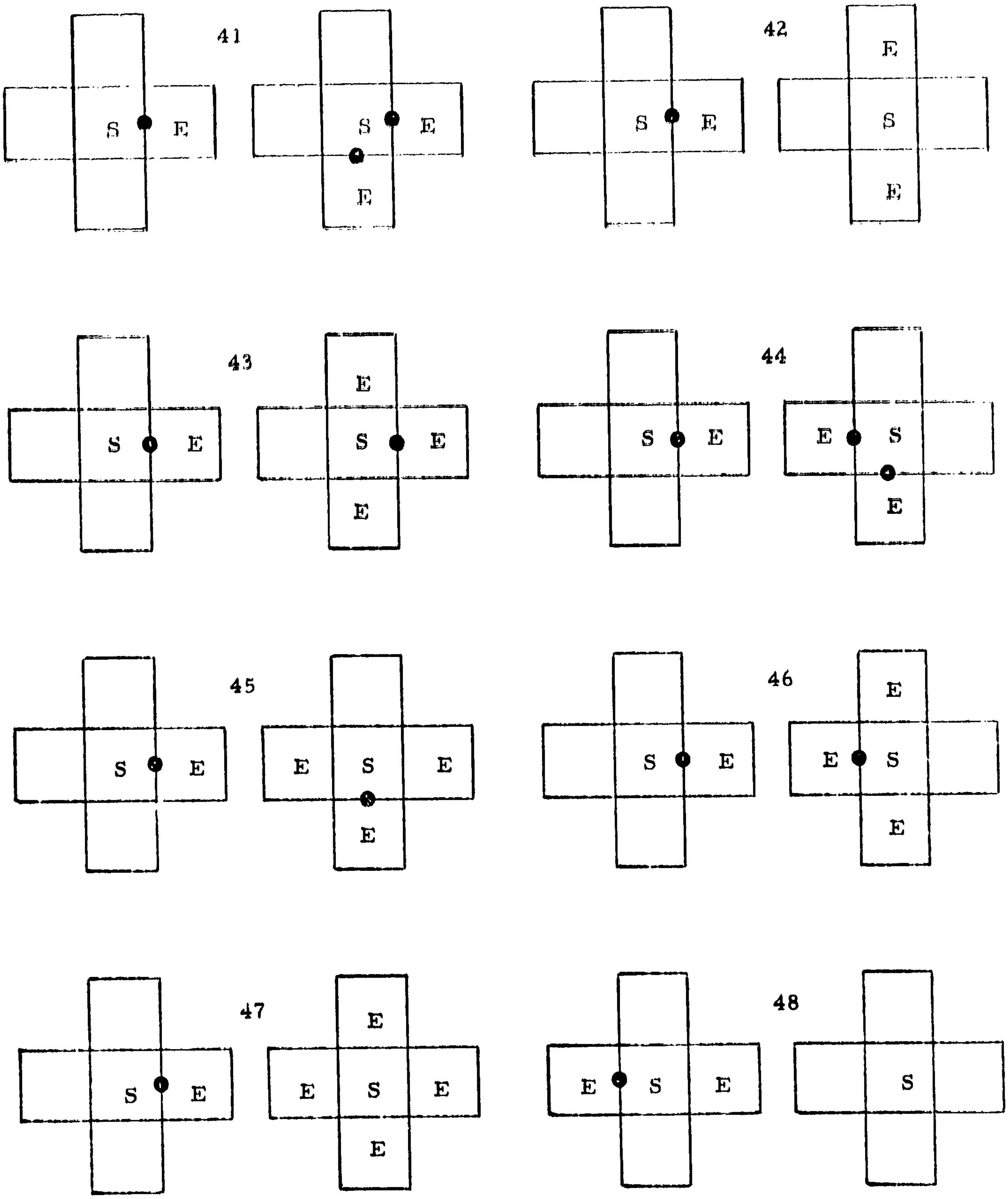


Fig. 2 - (Continued)

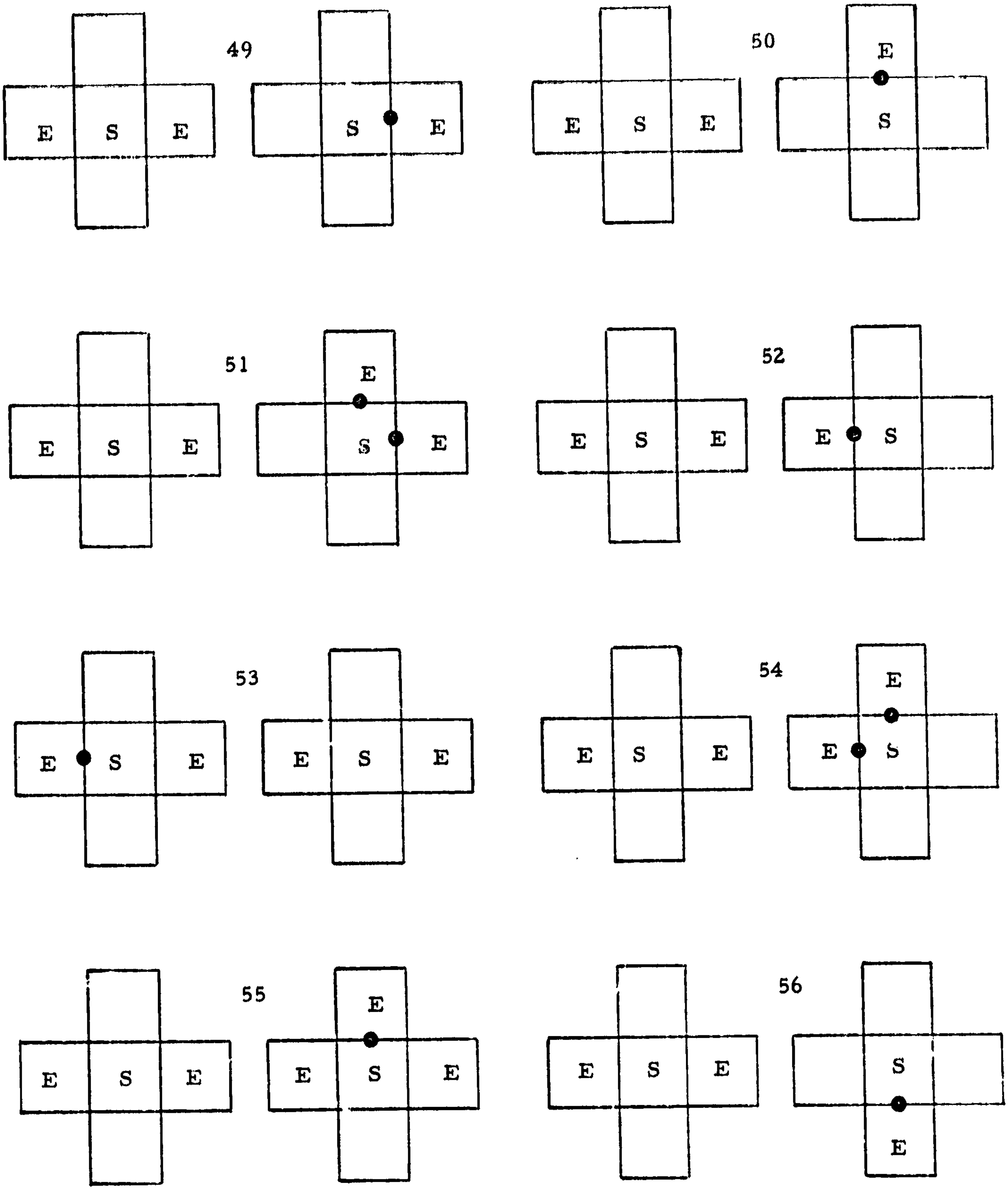


Fig. 2 - (Continued)

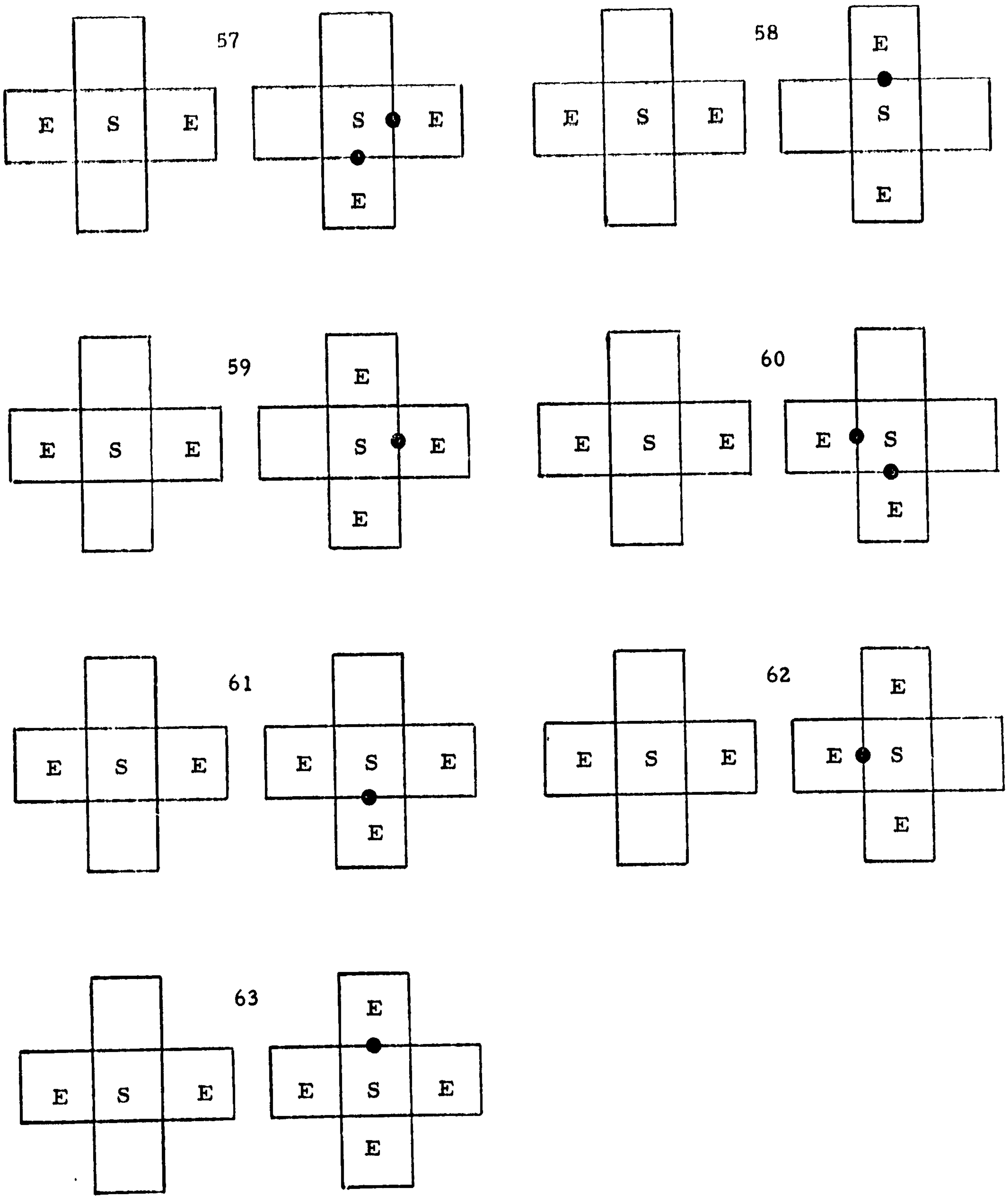


Fig. 2 - (Concluded)

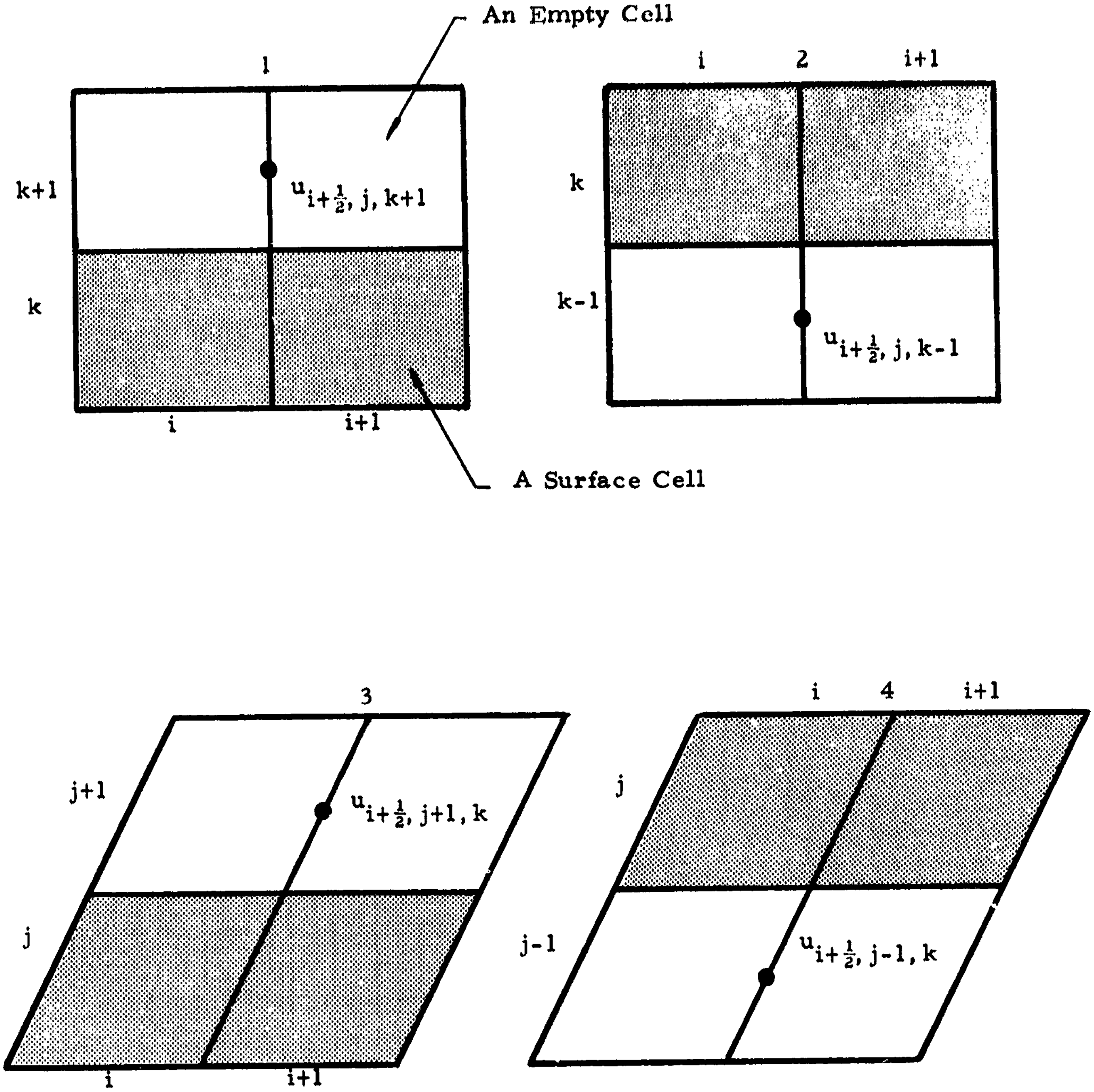


Fig. 3 – Twelve Possible Cases of Two Empty Cells Neighboring with a Free Surface

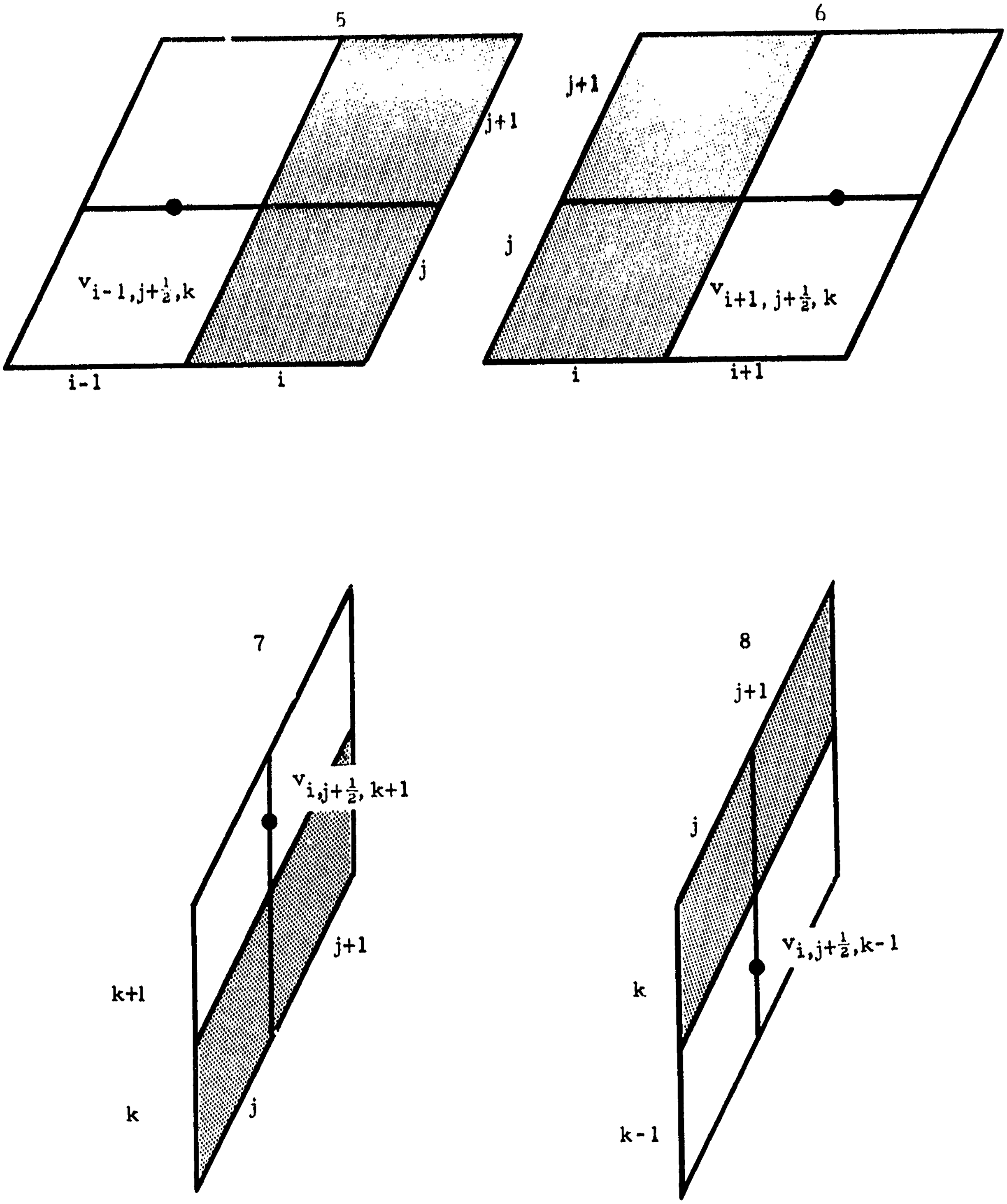

5
j+1
$v_{i-1,j+\frac{1}{2},k}$
j
i-1
i
6
j+1
j
$v_{i+1,j+\frac{1}{2},k}$
i
i+1
7
$v_{i,j+\frac{1}{2},k+1}$
j+1
k+1
k
j
8
j+1
j
$v_{i,j+\frac{1}{2},k-1}$
k
k-1

Fig. 3 (Cont'd)

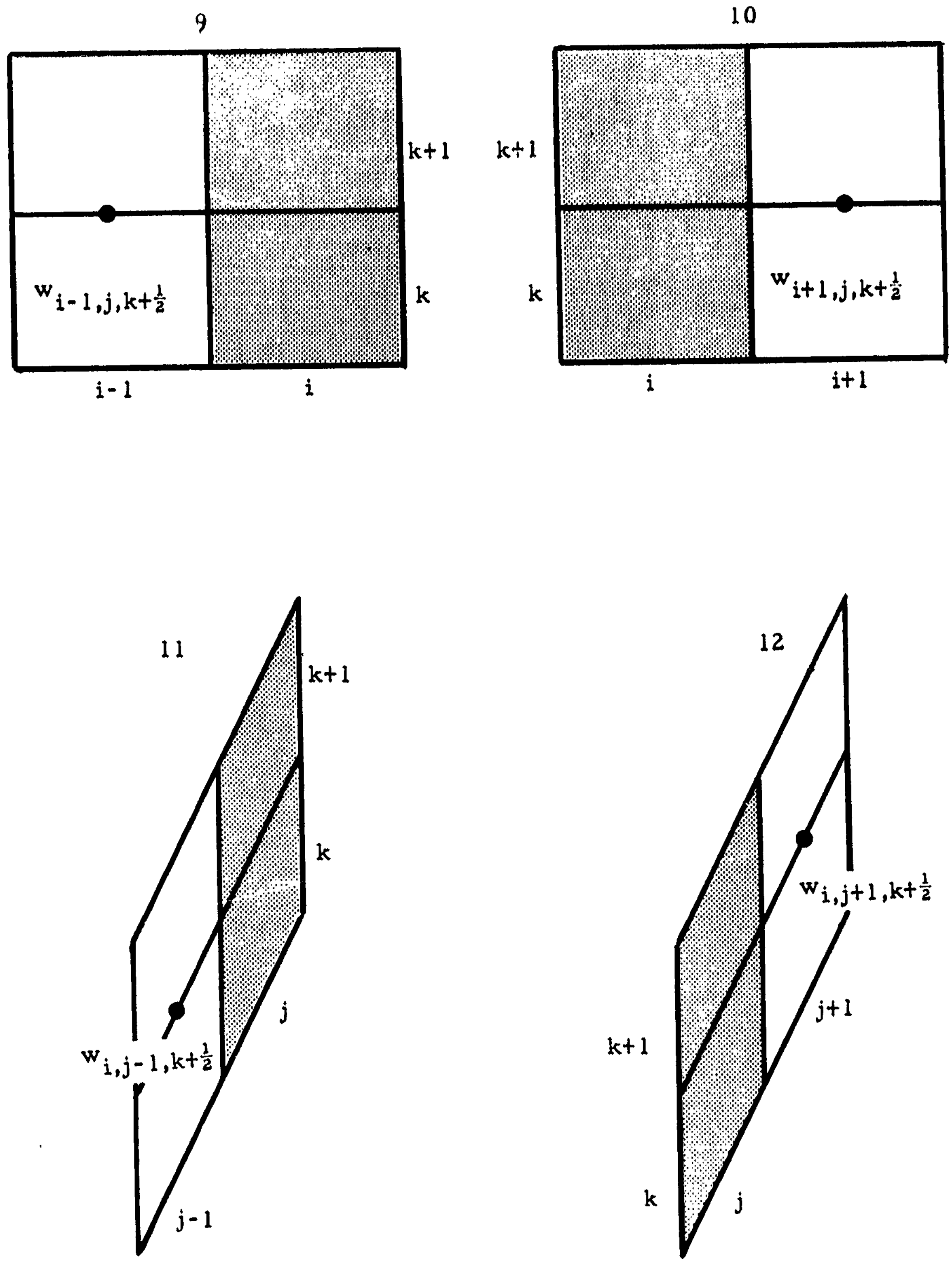


Fig. 3 - (Concluded)

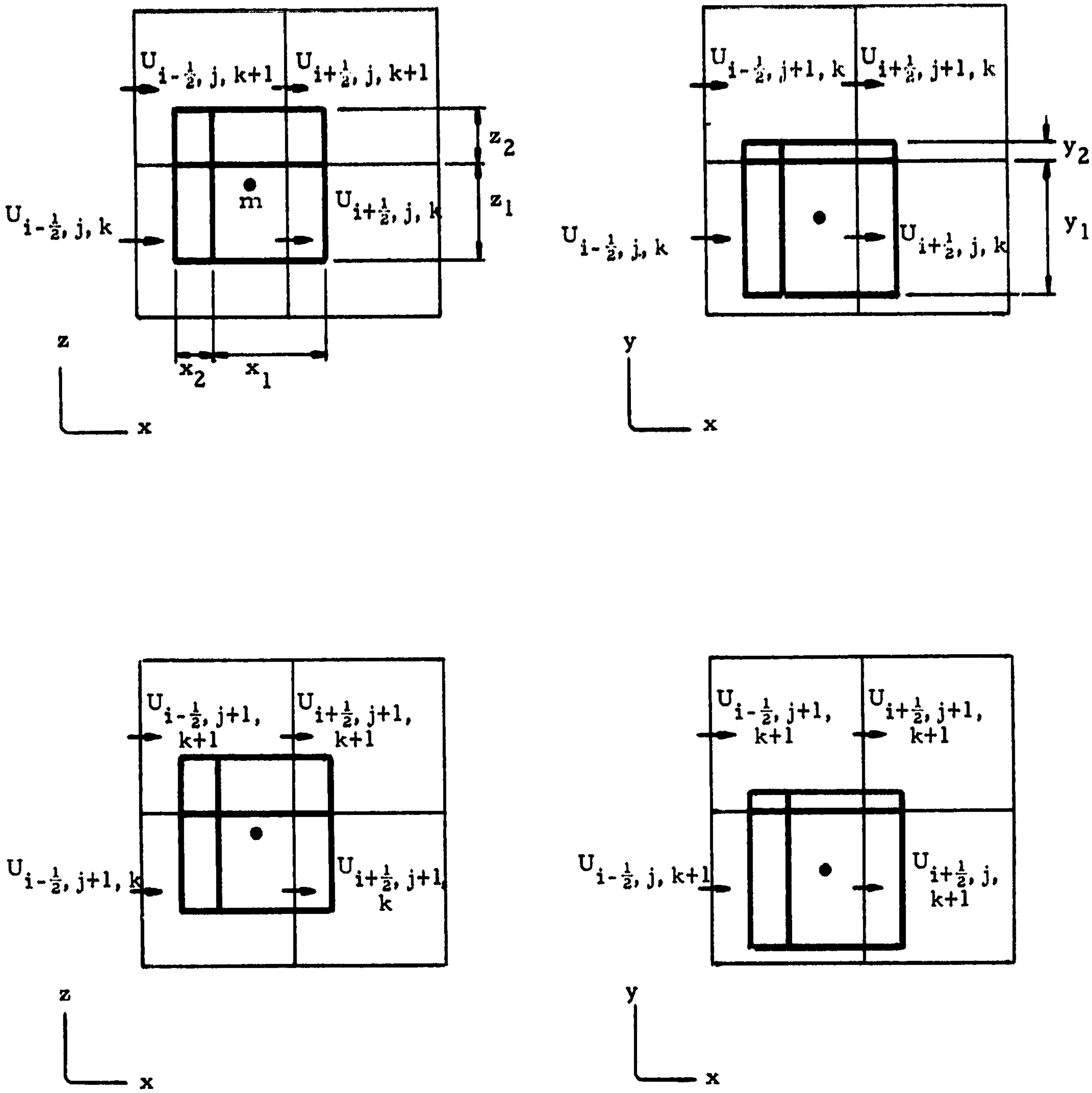


Fig. 4 - A Volume-Velocity Weighting Scheme for Calculating the Velocities of a Marker Particle in Cell (K, J, I)

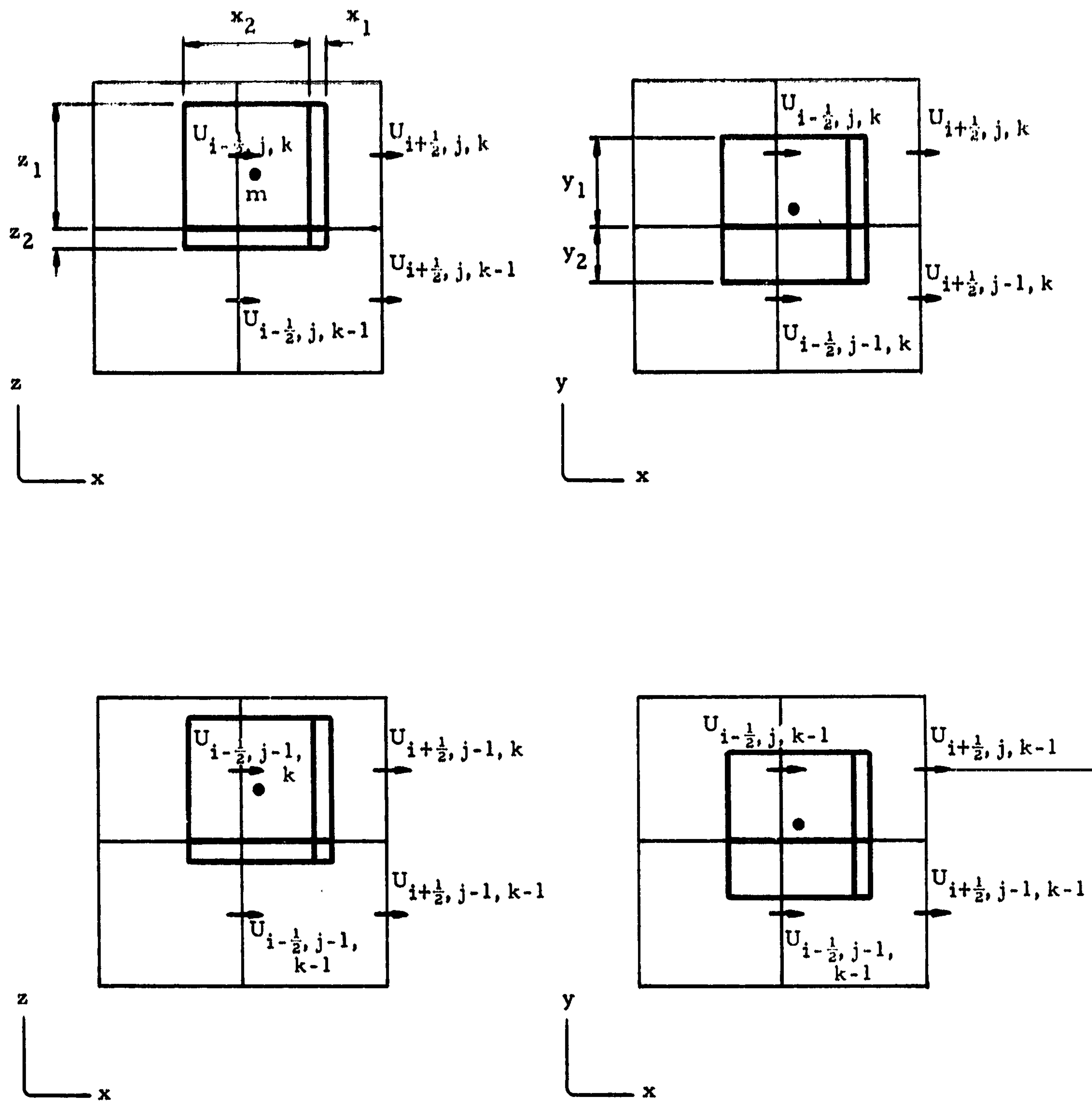


Fig. 4 - (Concluded)

NOTE: Length is measured as number of cells

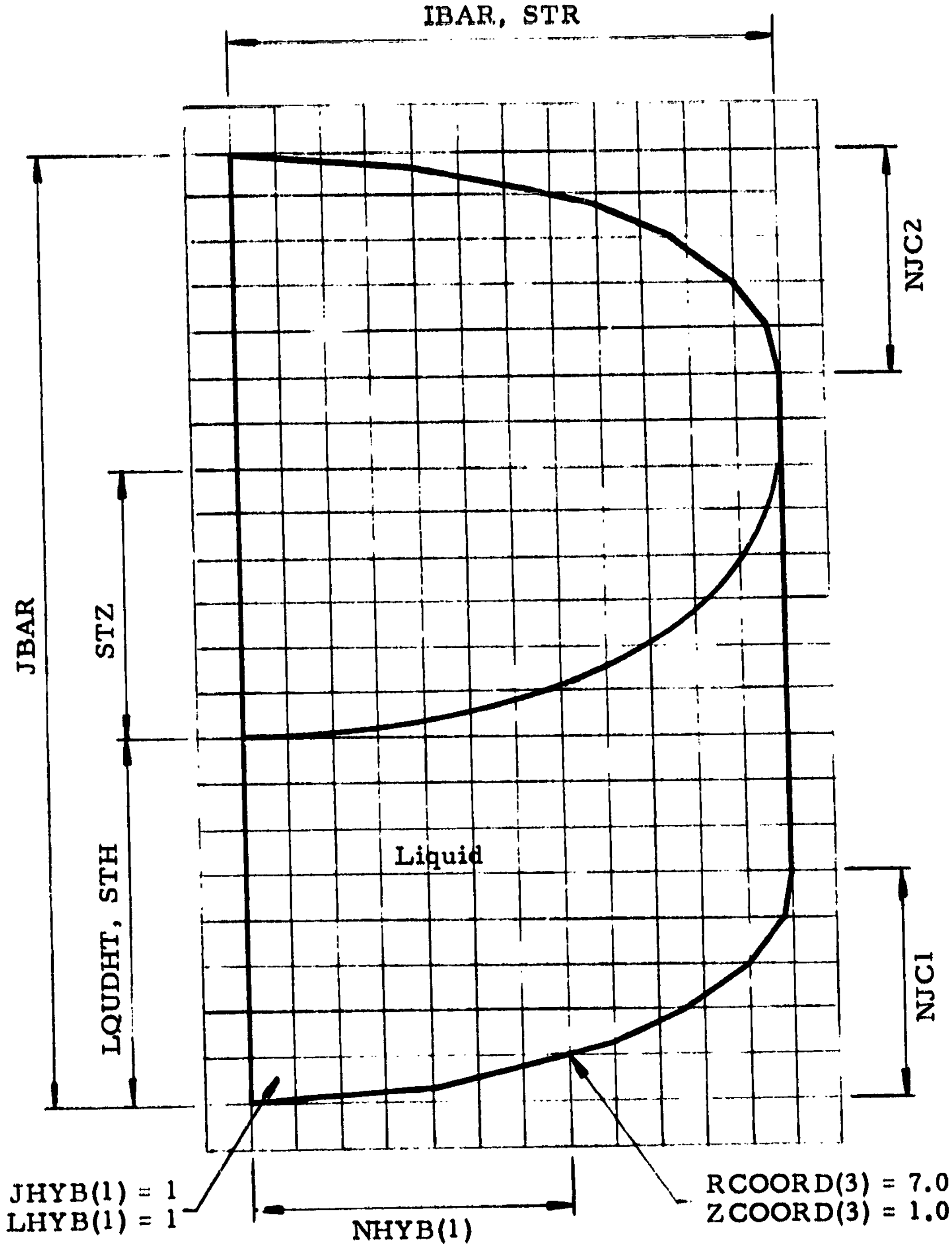


Fig. 5 - Variables to be Used in Preparing the Data Deck of the LHMAC2 Program

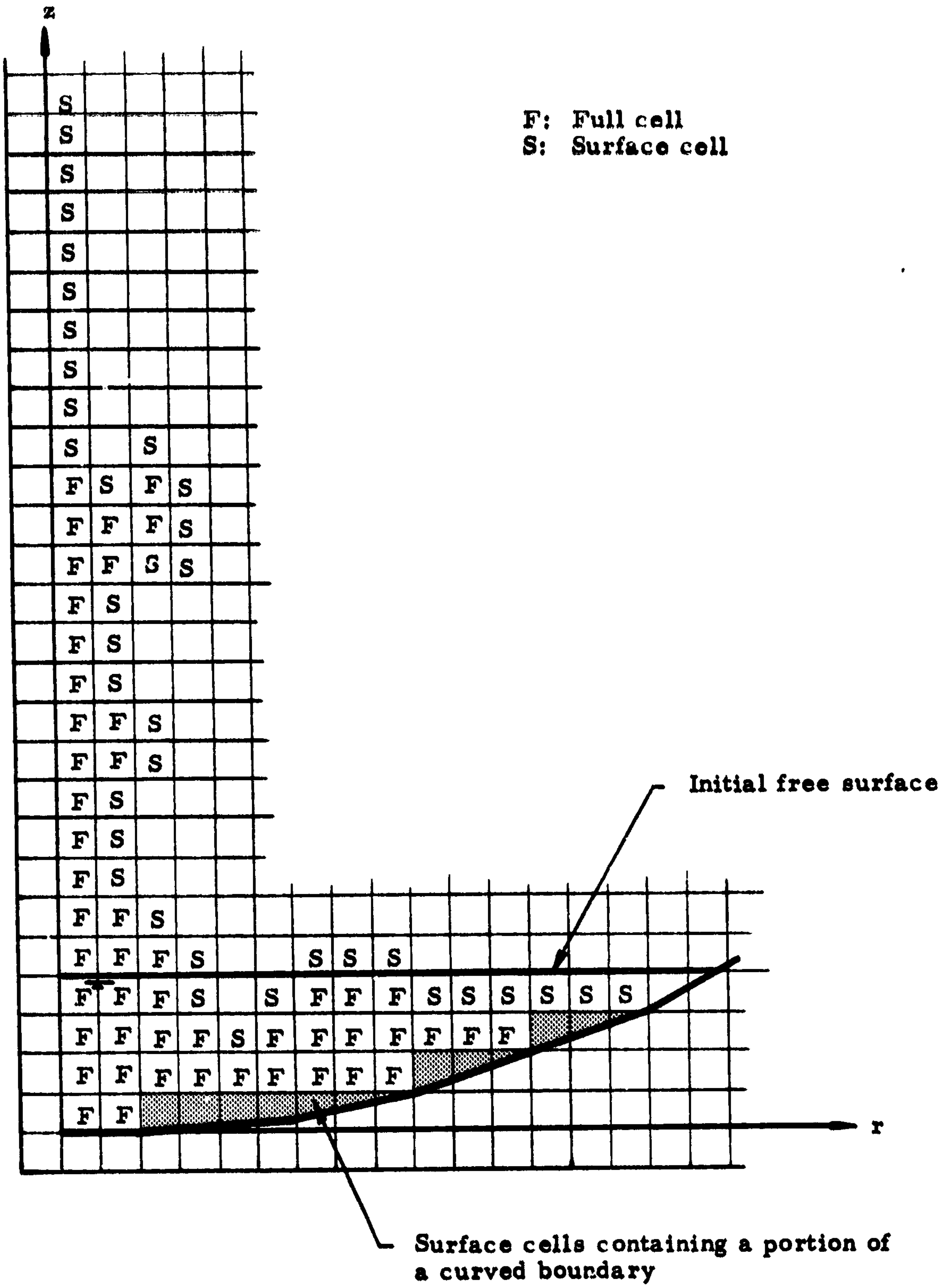


Fig. 6 - Cell Status of Sample Problem I at t = 1 sec

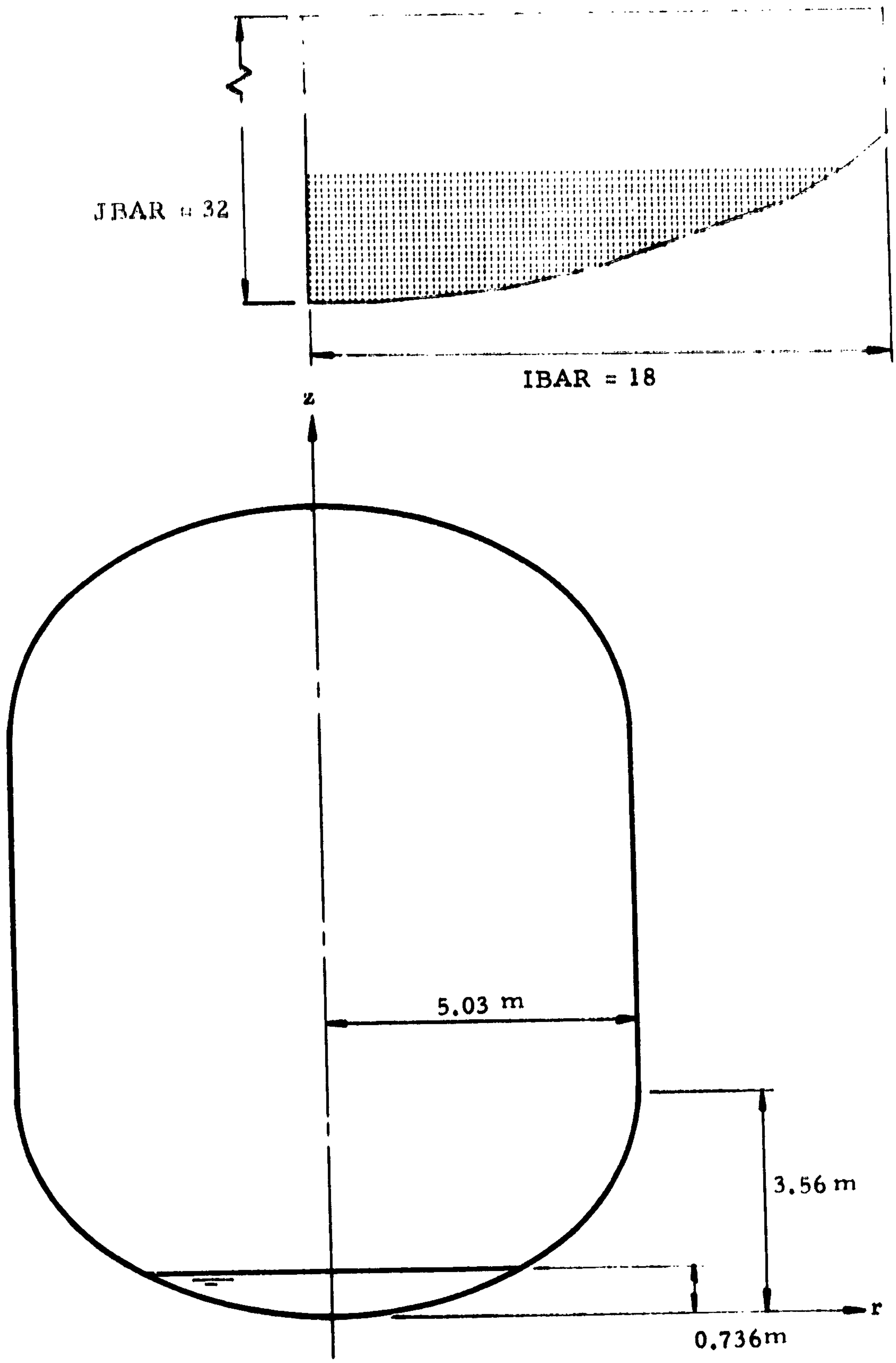
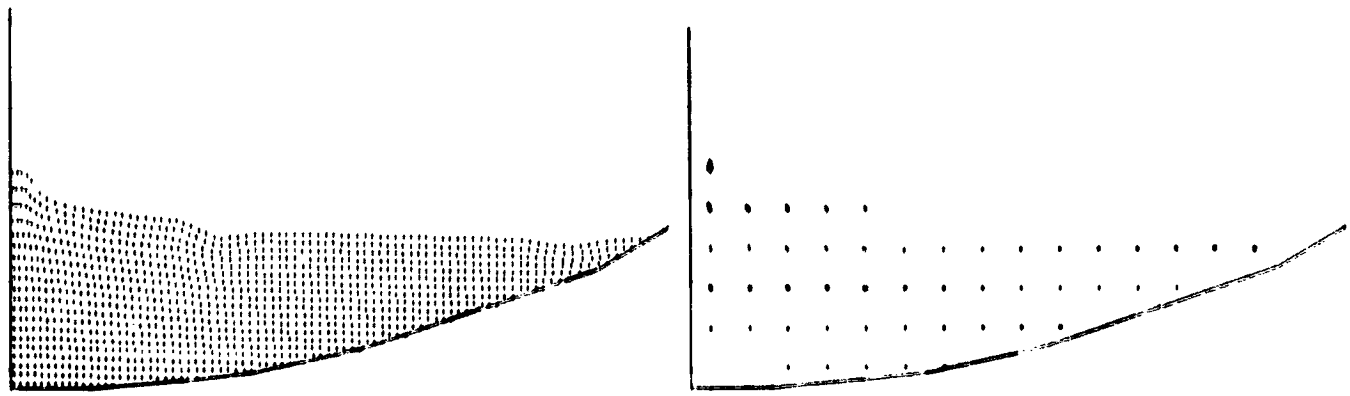
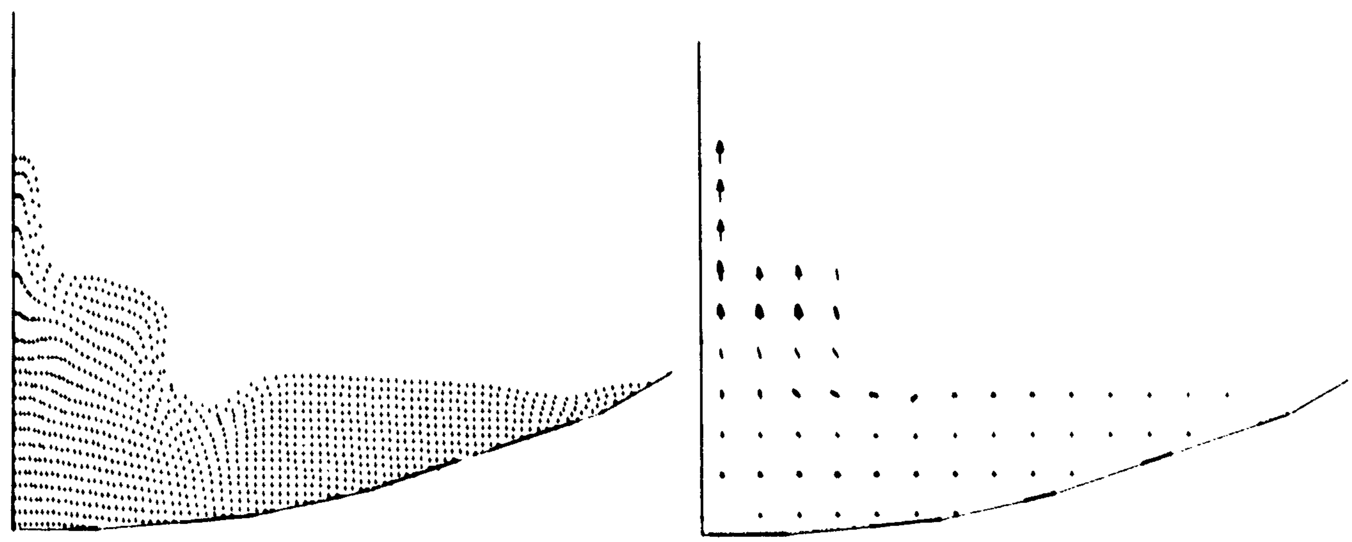


Fig. 7 - Sample Problem I – Container Geometry and Liquid Height

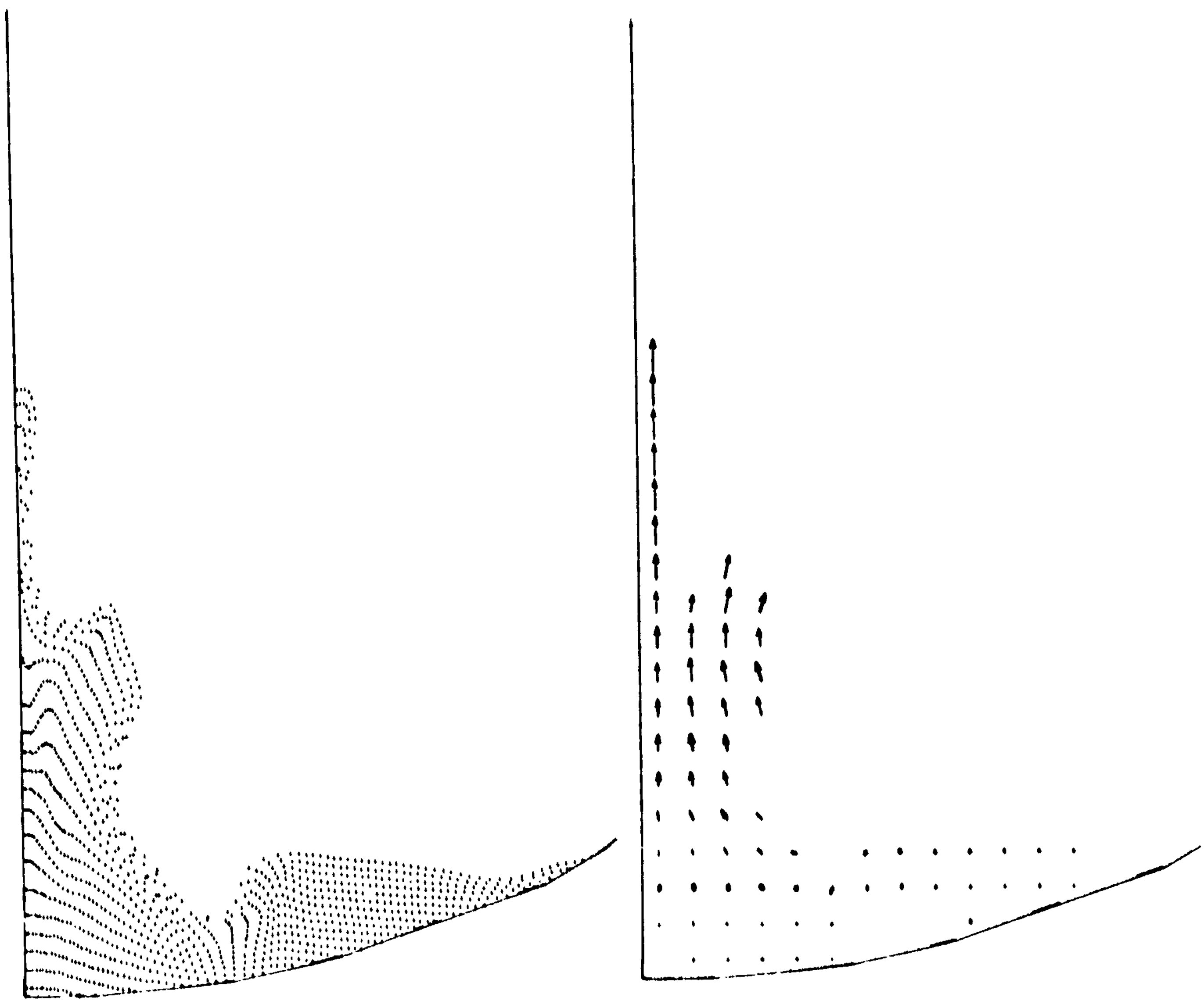


t = 0.25 sec



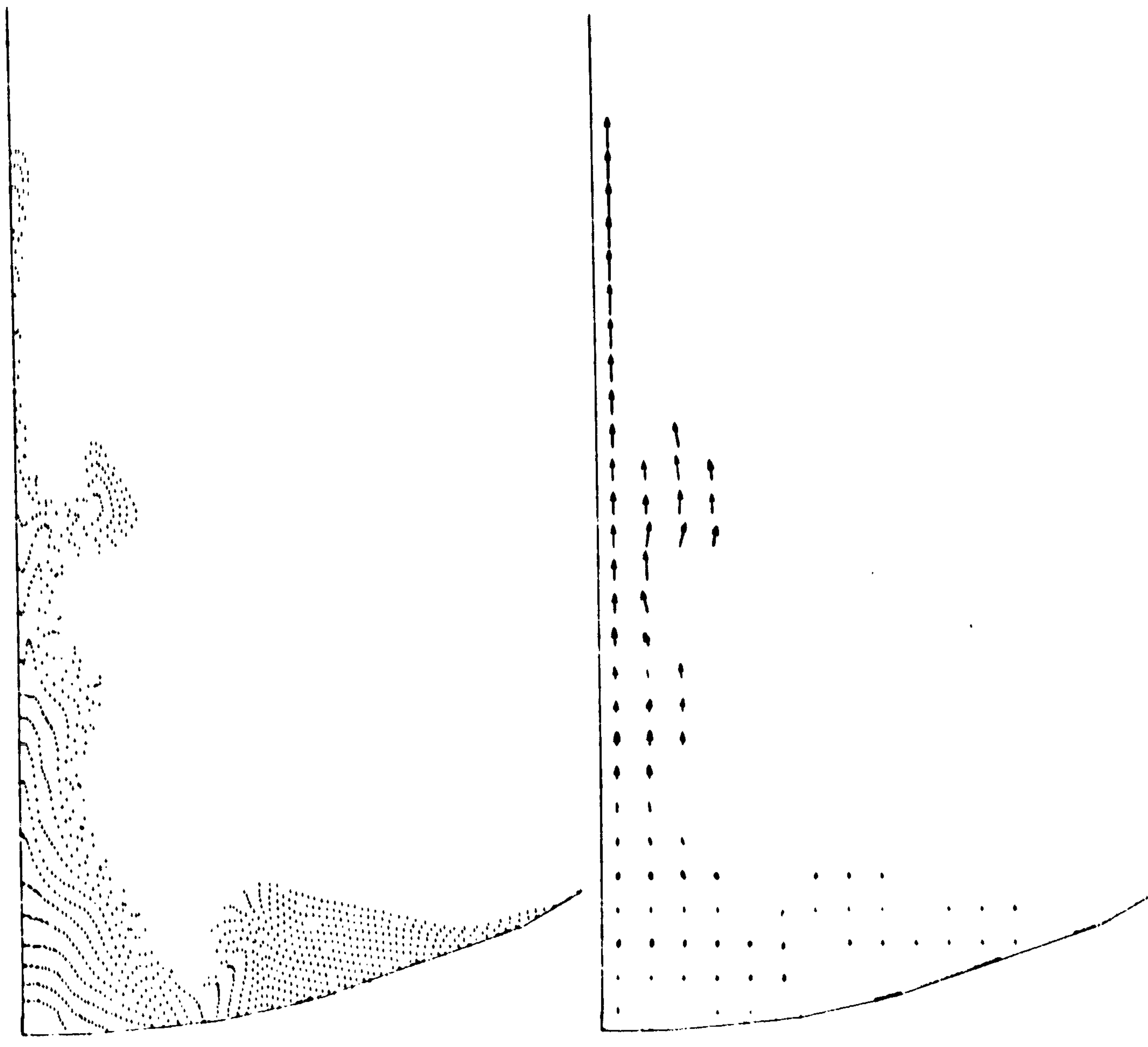
t = 0.5 sec

Fig. 8 - Sample Problem I – Flow and Velocity Fields at Selected Times



t = 0.75 sec

Fig. 8 - (Continued)



t = 1.0 sec

Fig. 8 - (Concluded)

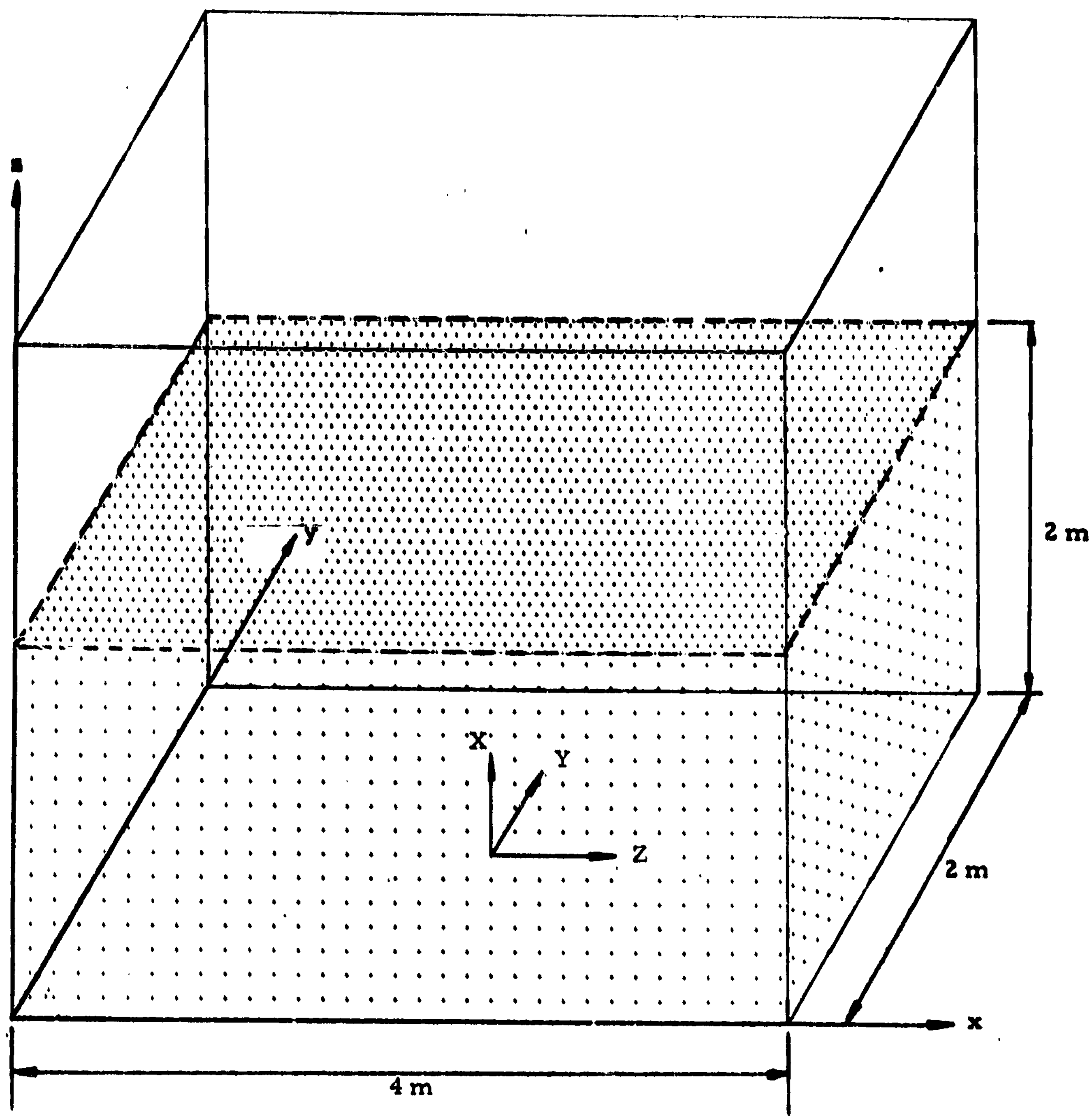
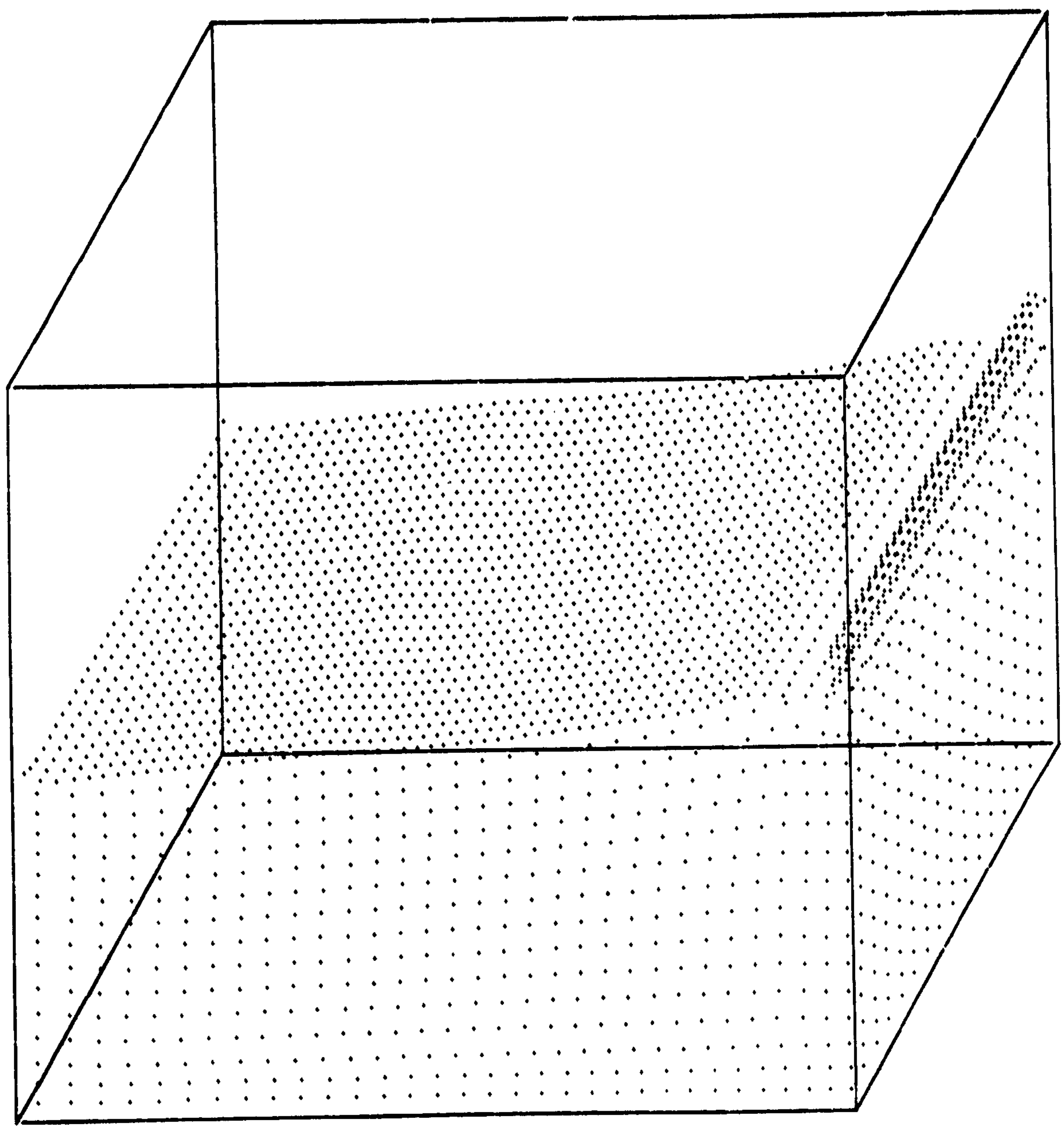
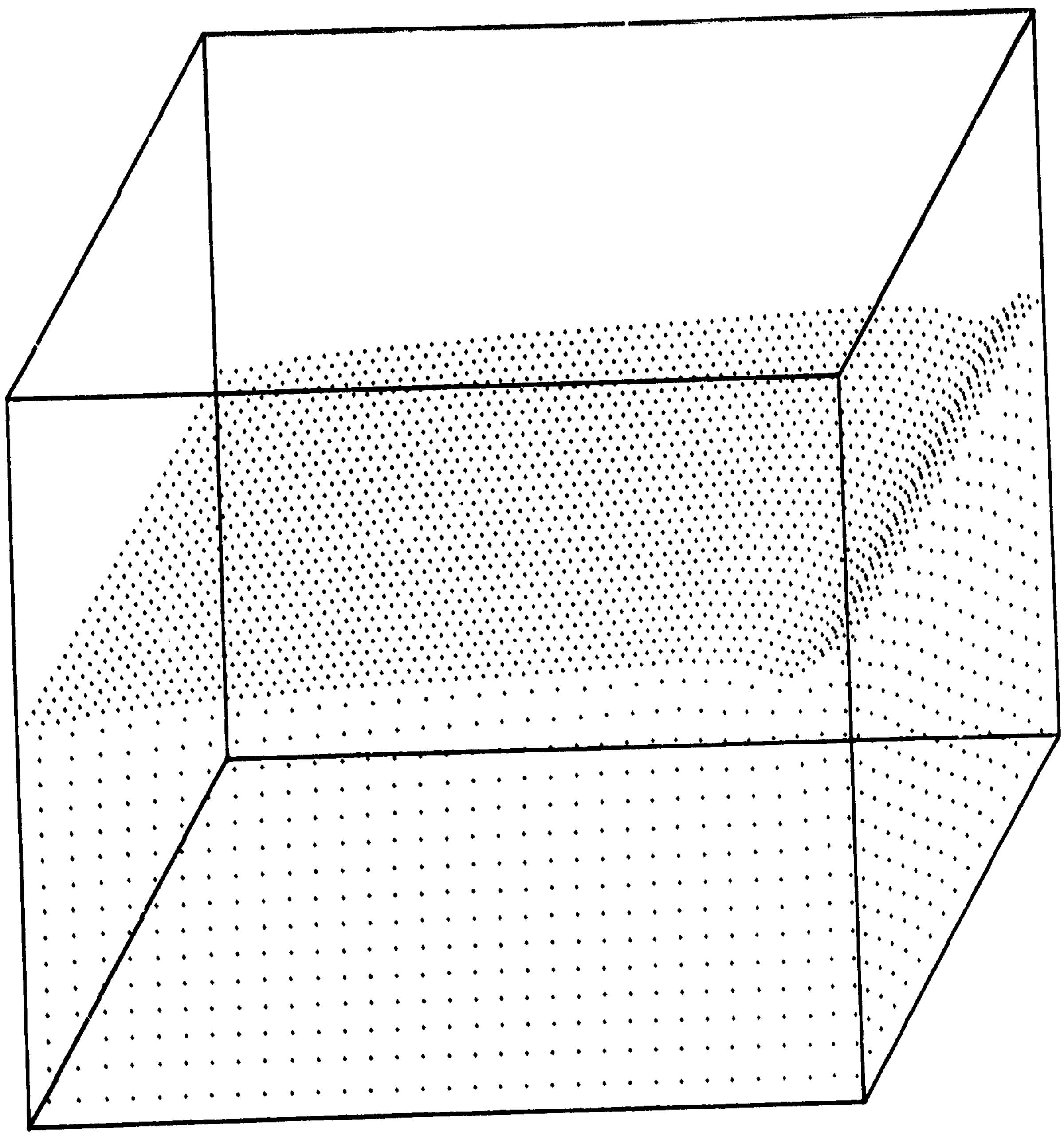


Fig. 9 - Sample Problem II – Container Geometry and Liquid Height



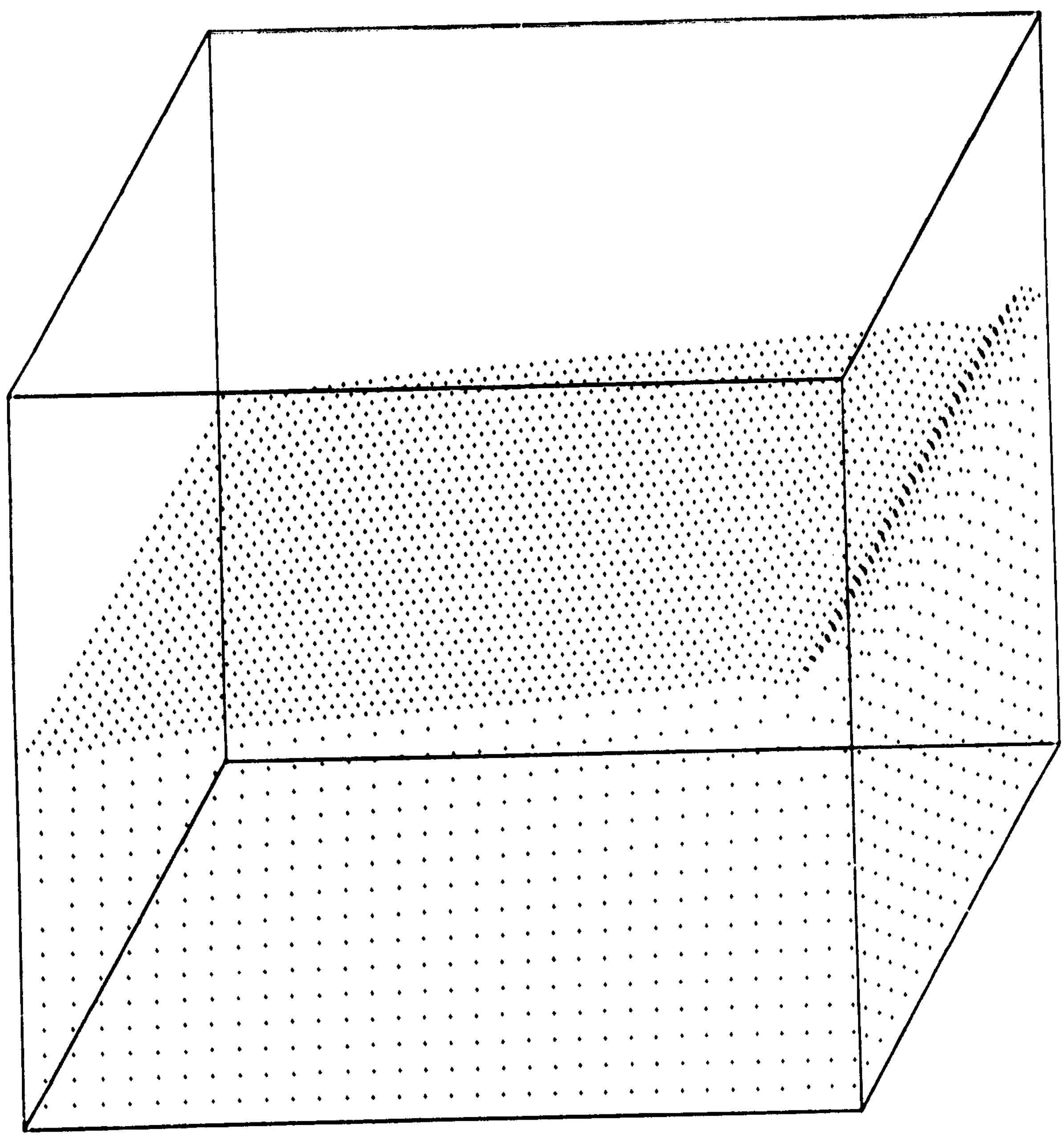
t = 0.5 sec (Case 1)

Fig. 10 - Sample Problem II – Flow and Velocity Fields at Selected Times



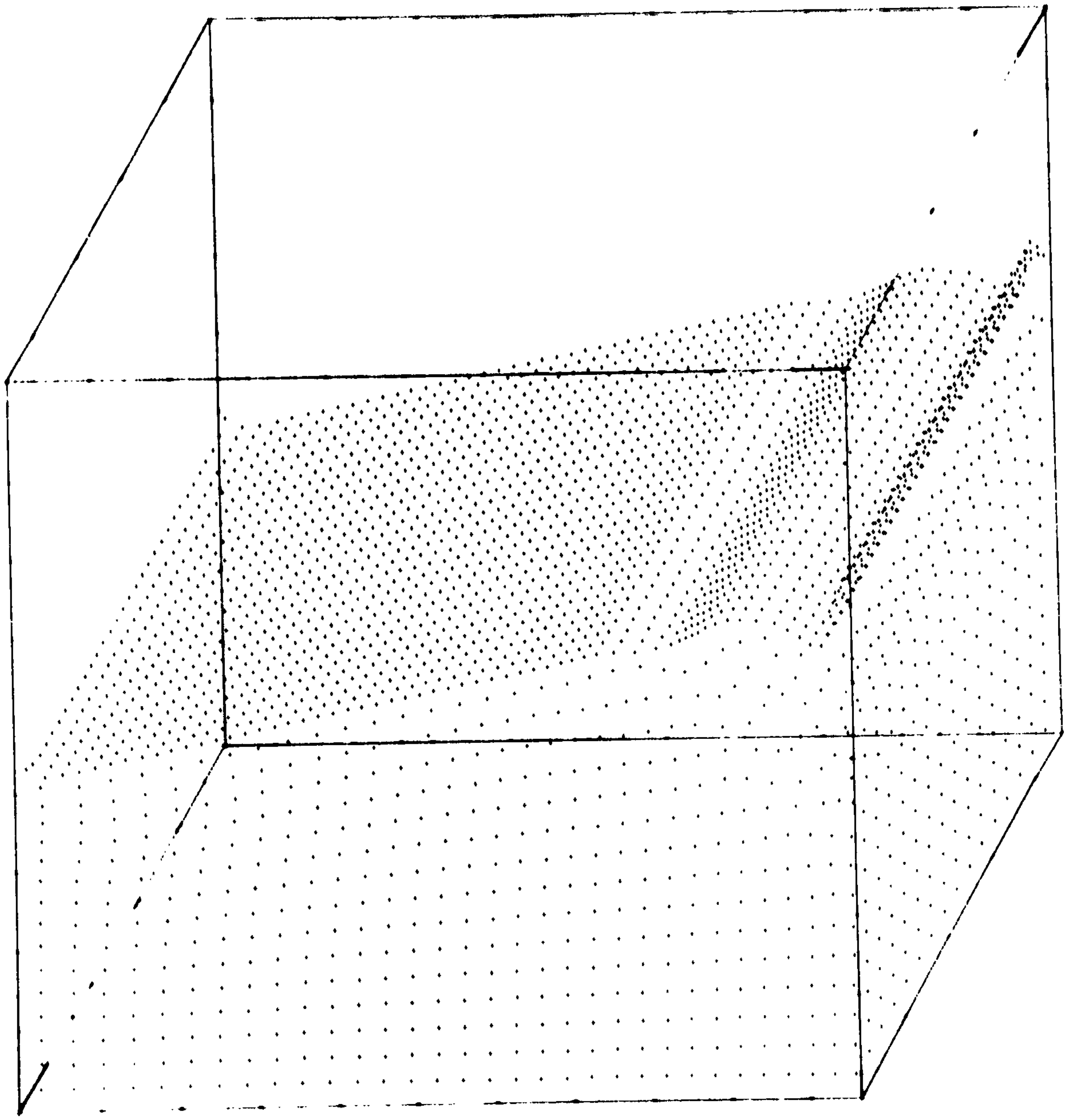
t = 0.3125 sec (Case 2)

Fig. 10 (Continued)



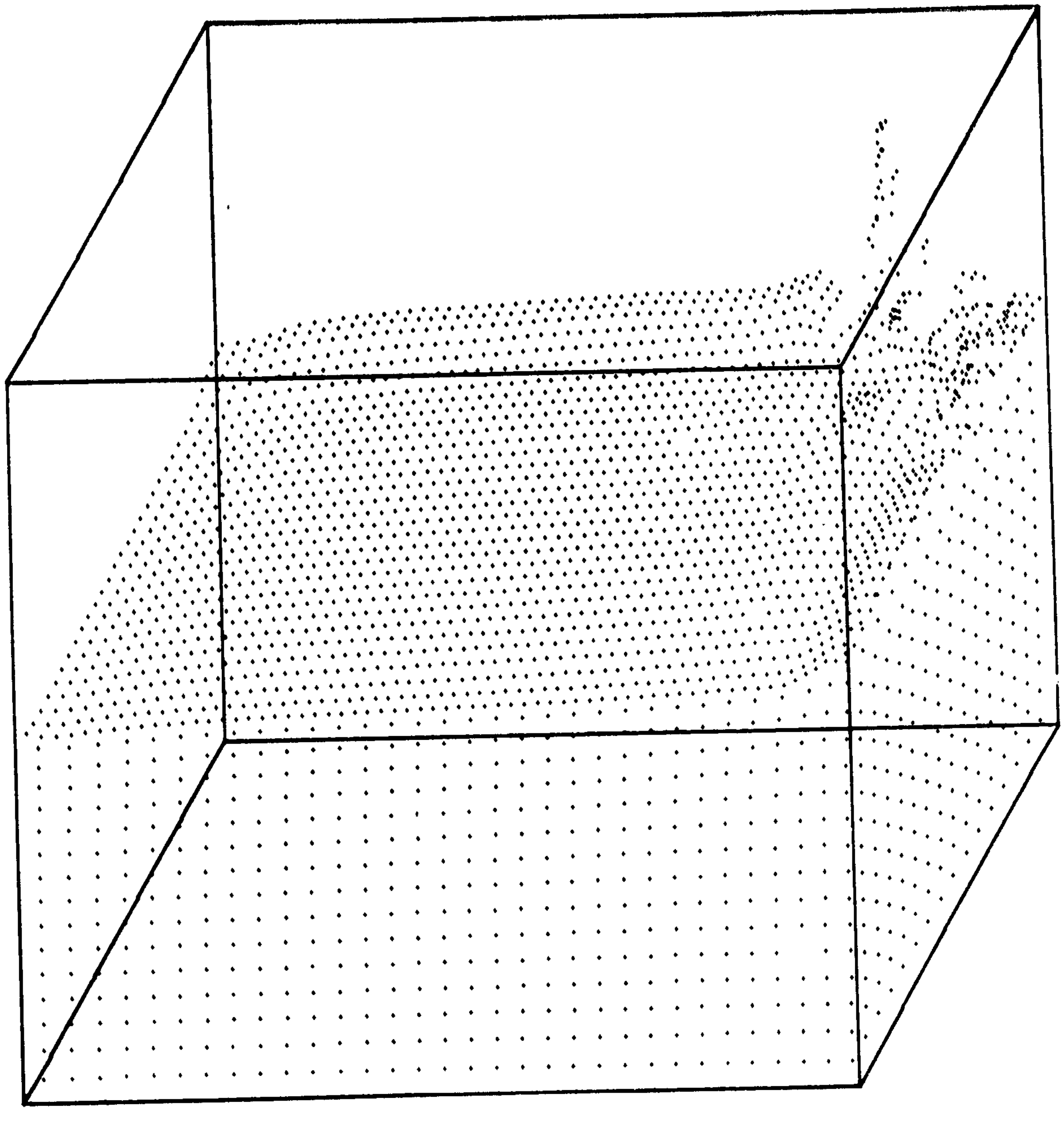
t = 0.5 sec (Case 3)

Fig. 10 - (Continued)



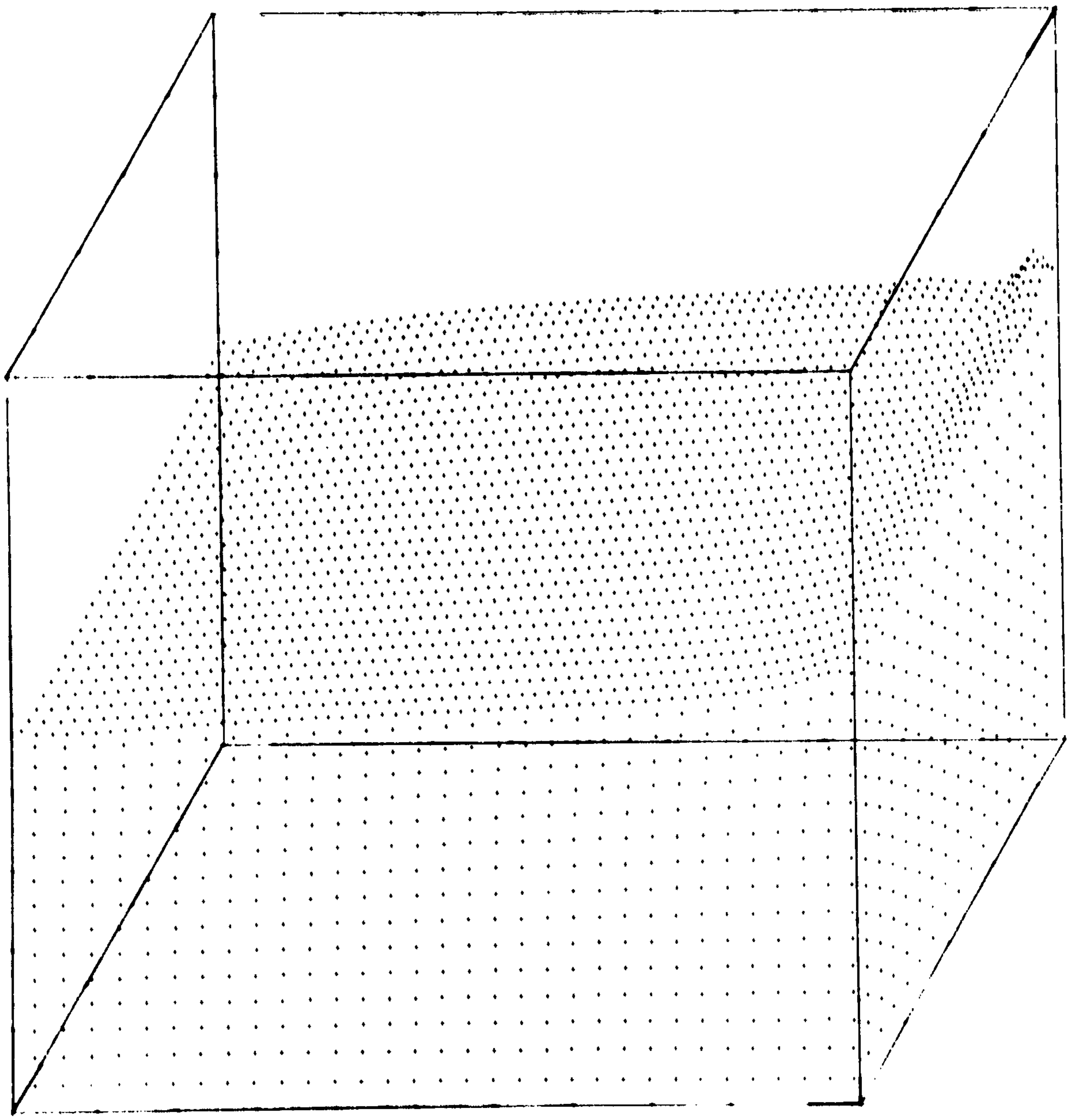
t = 0.875 sec (Case 3)

Fig. 10 - (Continued)



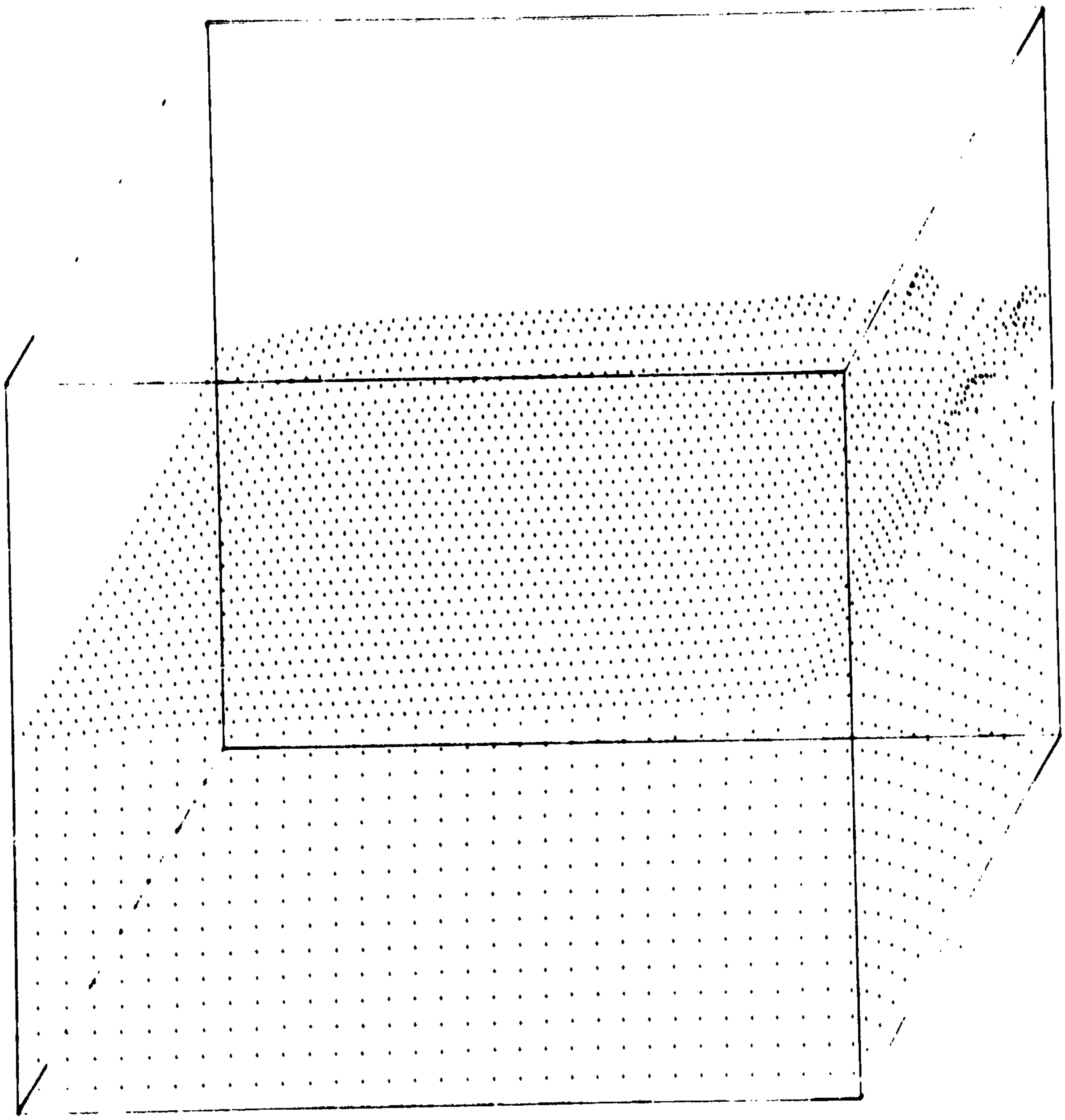
t = 0.34375 (Case 4)

Fig. 10 - (Continued)



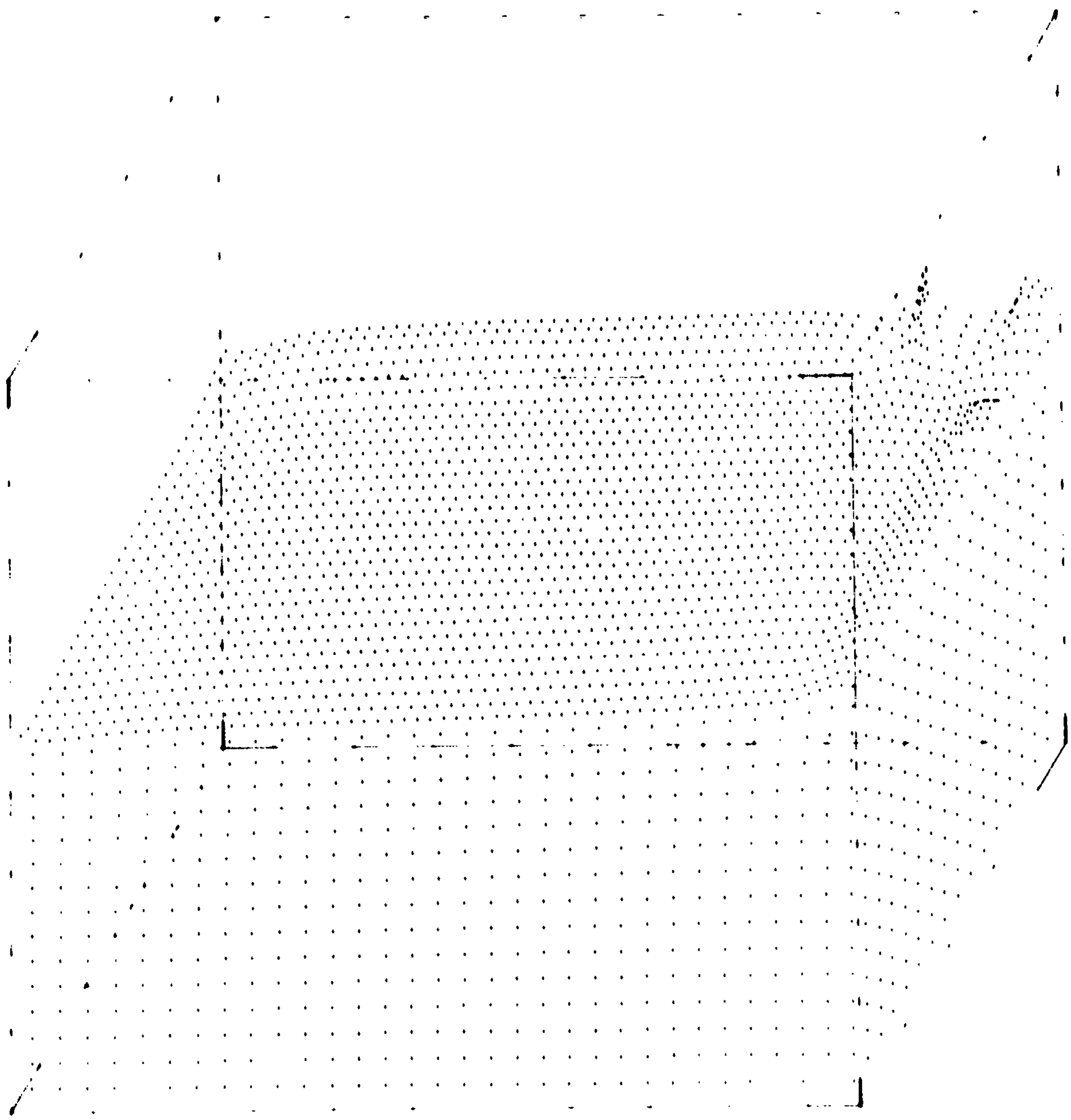
t = 0.34375 sec (Case 5)

Fig. 10 - (Continued)



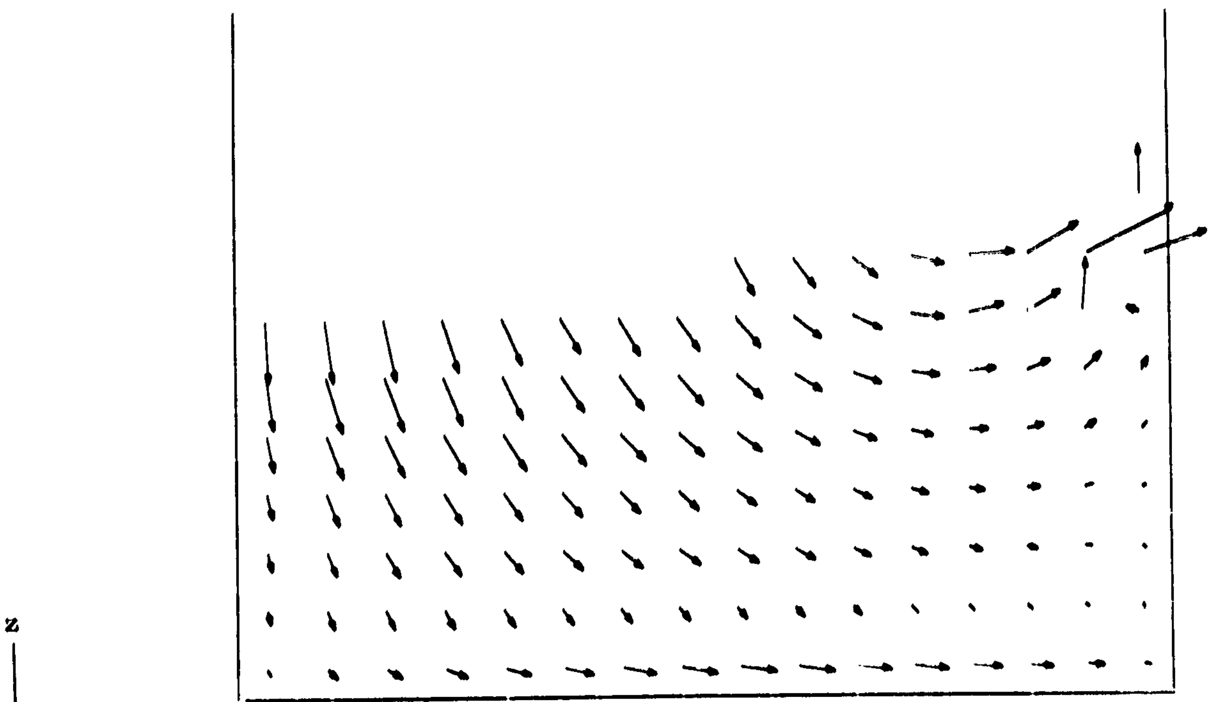
t = 0.3125 sec (Case 6)

Fig. 10 - (Continued)

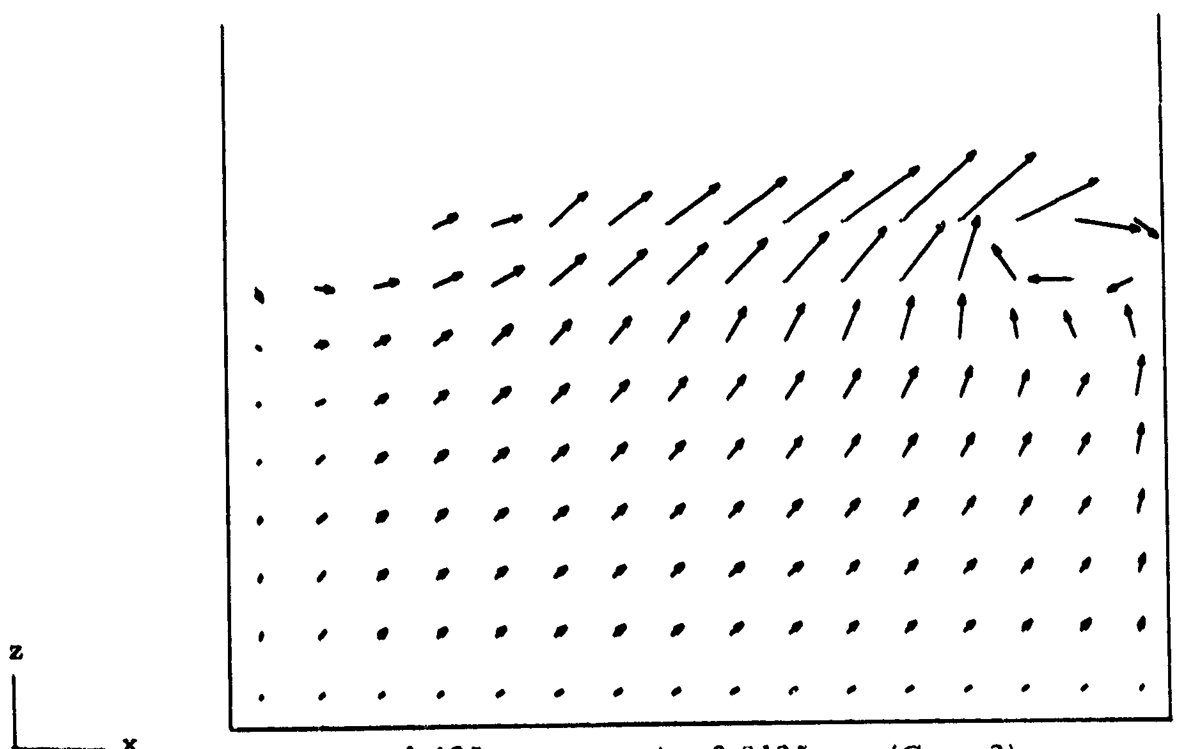


t = 0.375 sec (Case 7)

Fig. 10 - (Continued)

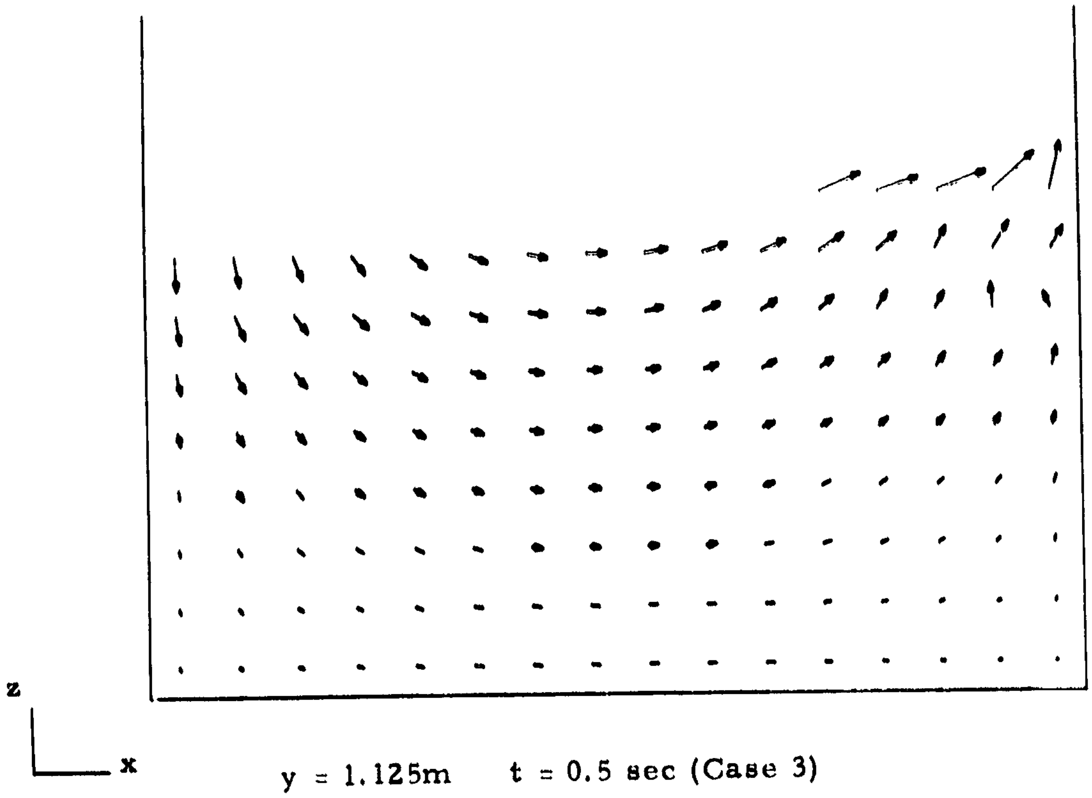


y = 1.125 m t = 0.5 sec (Case 1)

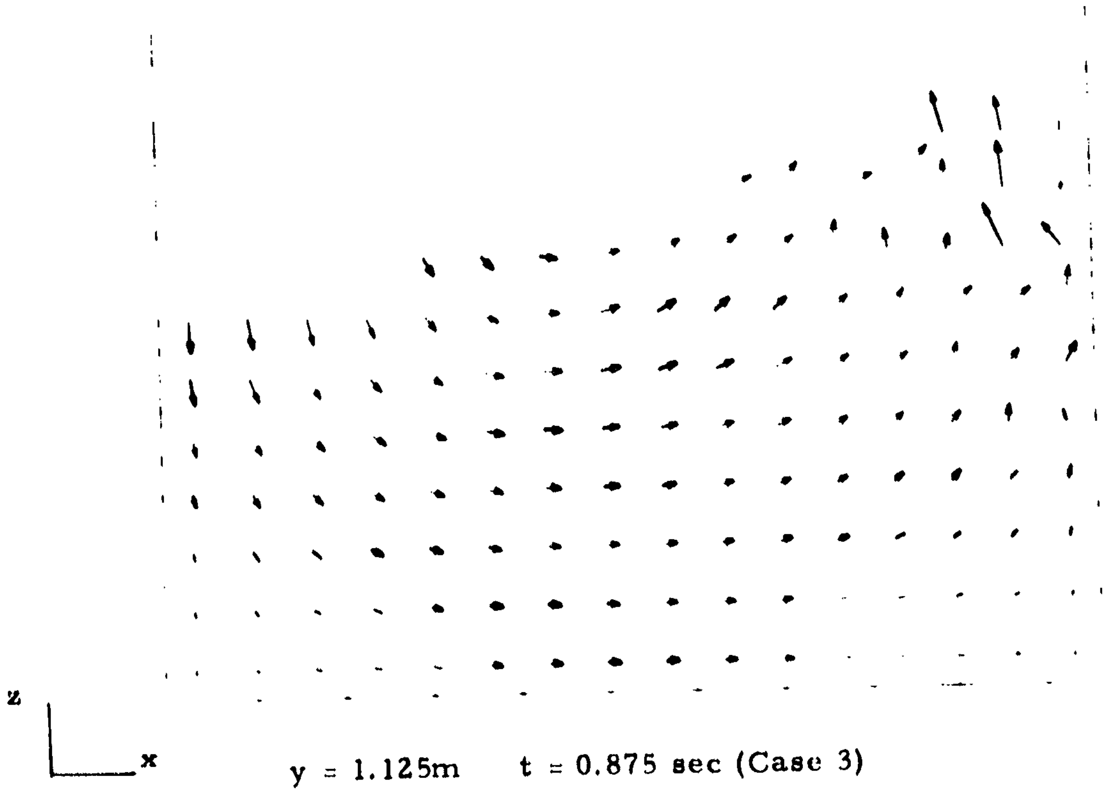


y = 1.125 m t = 0.3125 sec (Case 2)

Fig. 10 - (Continued)

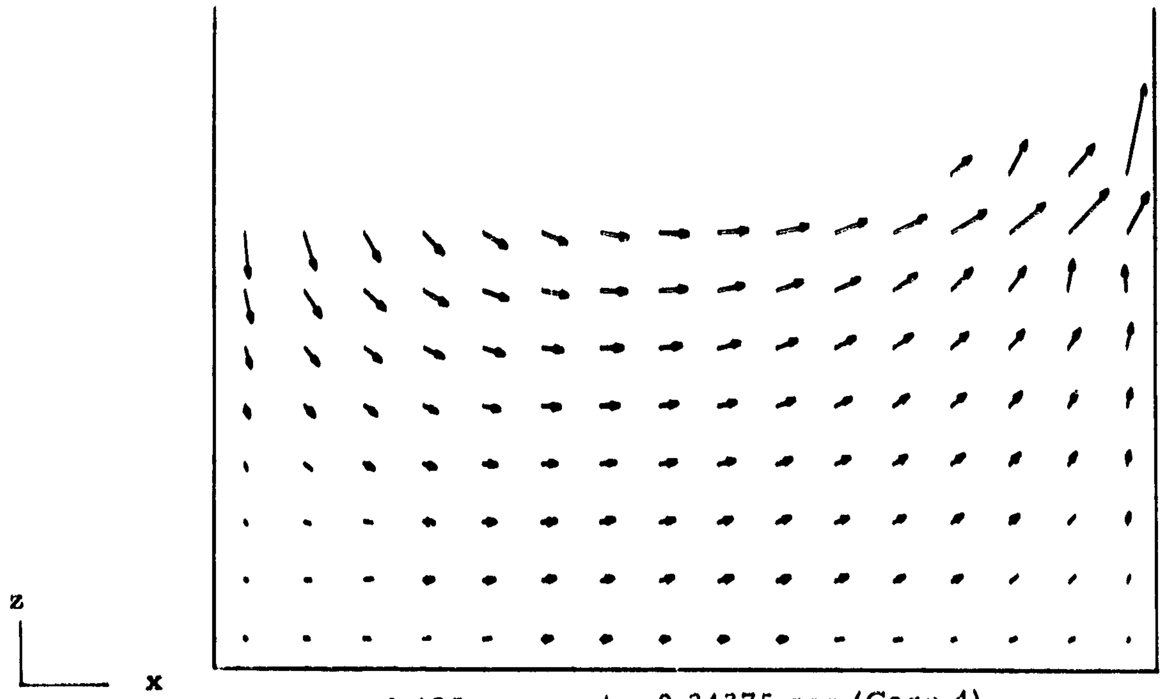


y = 1.125m   t = 0.5 sec (Case 3)

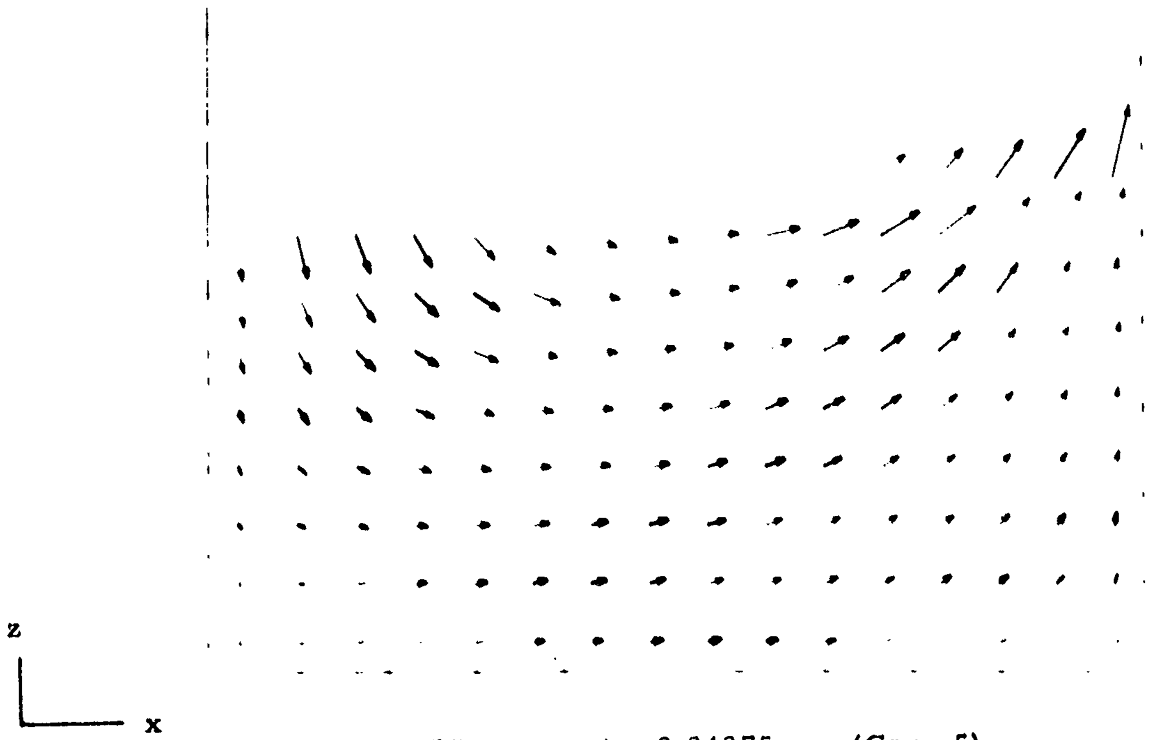


y = 1.125m   t = 0.875 sec (Case 3)

Fig. 10 (Continued)

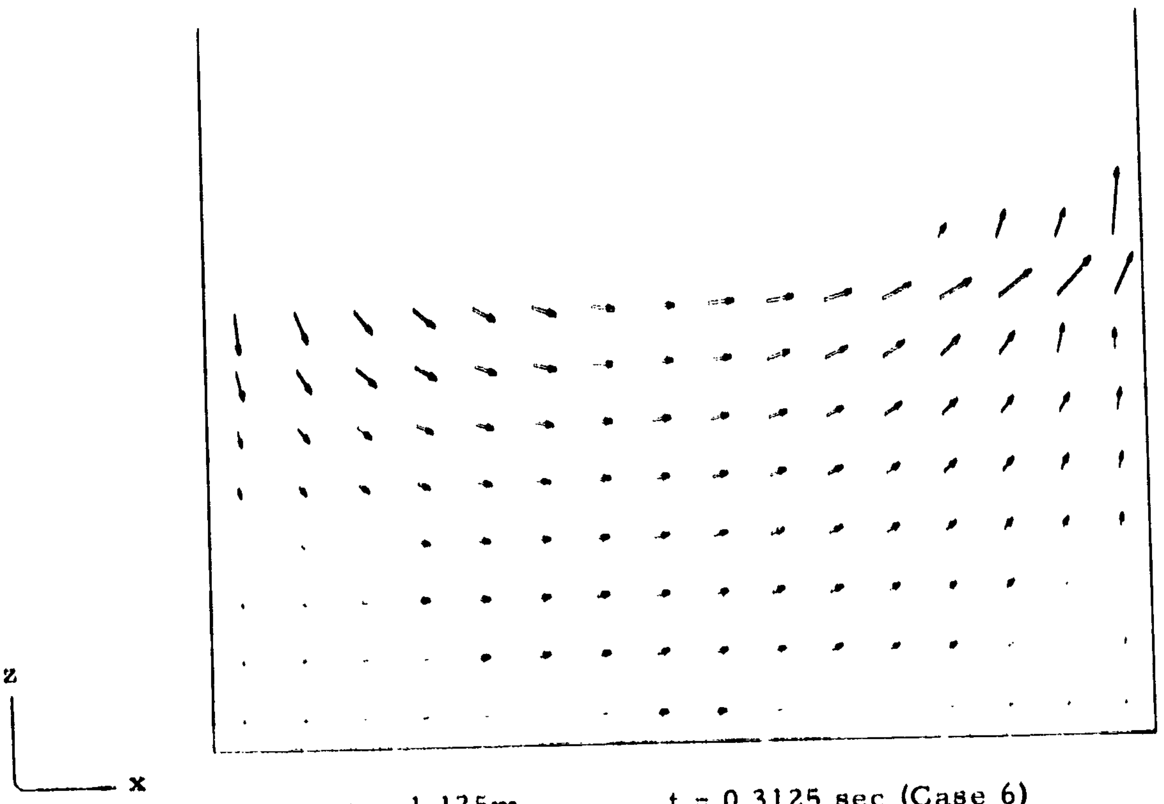


y = 1.125m t = 0.34375 sec (Case 4)

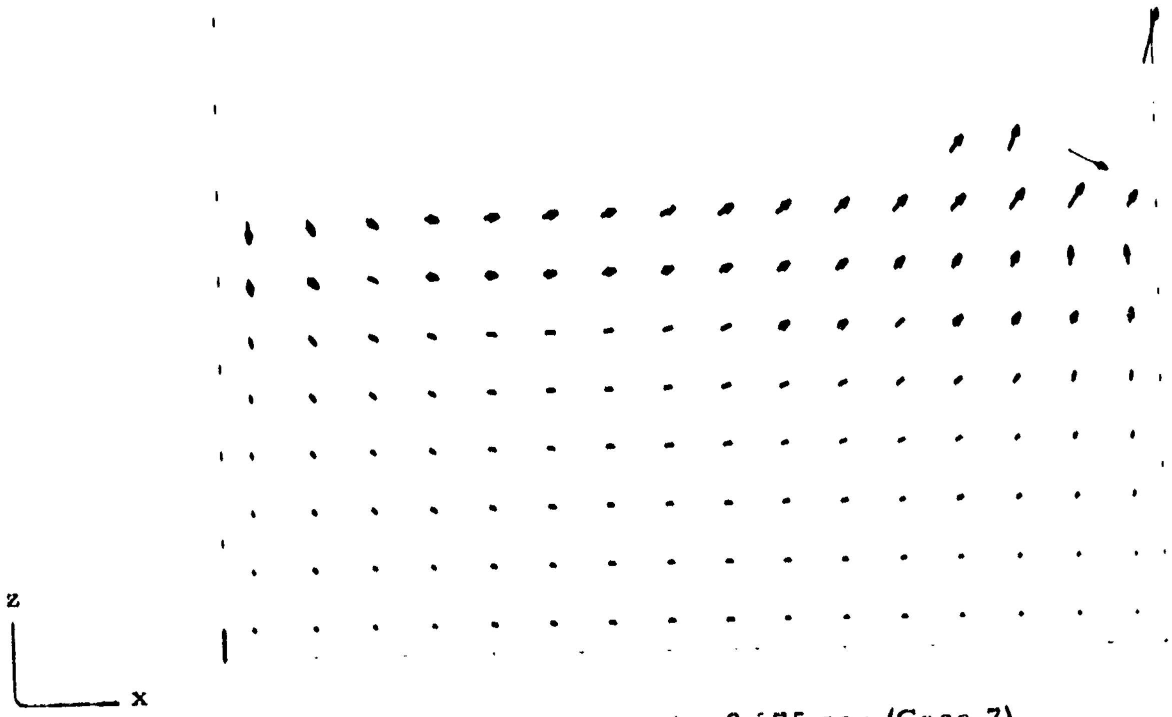


y = 1.125m t = 0.34375 sec (Case 5)

Fig. 10 - (Continued)



y = 1.125m t = 0.3125 sec (Case 6)



y = 1.625m t = 0.375 sec (Case 7)

Fig. 10 - (Concluded)

Appendix A

LISTING OF LHMAC2 PROGRAM

Listing of LHMAC2 Program

```
@RUN,//T
@MSG,N LOCKHEED-HUNTSVILLE 2D MAC PROGRAM (TAPE GEN 090272)
@ASG,T LHMAC2.T.SAVE05 . LOCKHEEDMAC2PROGRAM
@REWIND LHMAC2.
@FOR,IS 801,801
C
C LOCKHEED/HUNTSVILLE 2D MAC PROGRAM (LHMAC2, 56K CORE SPACE)
C
      COMMON/L1/ ALP, DR, DZ, DT, IBAR, IPHM, JBAR, PC,
     1  A1(1200), A2(1200), A3(1200), A4(1200), A5(1200), A6(1200),
     2  A7(1200), A8(1200), A9(1200), A10(1200), A11(1200), A12(1200),
     3  A13(1200)
      EQUIVALENCE (A1,F),(A2,U),(A3,V),(A4,UTIL),(A5,VTIL),(A6,PSI),
     * (A7,THETA),(A8,D),(A9,KF),(A10,G),(A11,H),(A12,P),(A13,Q)
      INTEGER A1, A9, F, PC
      INTEGER TYPE(22)
      DATA TYPE/12H 9 IN PAPER , 20*6H      /
    1 FORMAT(2I5, 3F8.3,2I5,  F8.3)
      CALL IDENT(0,TYPE)
   10 READ  1, IBAR, JBAR, DR, DZ, DT, IPHM, PC, ALP
      IP2=IBAR+2
      JP2=JBAR+2
      CALL CLKOUT
      IF(IBAR) 30,20,20
   20 CALL PROG(IP2,JP2,F,U,V,UTIL,VTIL,PSI,THETA,D,KF,G,H,P,Q)
      GO TO 10
   30 CALL ENDJOB
      STOP
      END
@FOR,IS S02,S02
      SUBROUTINE PROG(IP2,JP2,F,U,V,UTIL,VTIL,PSI,THETA,D,KF,G,H,P,Q)
      COMMON/L1/ ALP, DR, DZ, DT, IBAR, IPHM, JBAR, PC,
     1  A1(1200), A2(1200), A3(1200), A4(1200), A5(1200), A6(1200),
     2  A7(1200), A8(1200), A9(1200), A10(1200), A11(1200), A12(1200),
     3  A13(1200)
      DIMENSION F(IP2,JP2),U(IP2,JP2),V(IP2,JP2),UTIL(IP2,JP2),
     1 VTIL(IP2,JP2),PSI(IP2,JP2),THETA(IP2,JP2),D(IP2,JP2),KF(IP2,JP2),
     2  G(IP2,JP2), H(IP2,JP2), P(IP2,JP2), Q(IP2,JP2)
      DIMENSION BCLT(60), BCRT(60), NAME(12),R(66), RIP(66), RMORP(66),
     1  ROS(64), RPORM(66), RRI(66),RRP(66)
      DIMENSION GAMMA( 80), XBDRY( 80), YBDRY( 80), ALPHA(80), DL(80),
     1  DB(128), FN(128), FX(128), FY(128)
      DIMENSION ISEGRR(35), ISEGRZ(35), IDUMP(2), JHYB(30), LHYB(29),
     1  NHYB(2), VN(80), IEMP(20), JEMP(20), GLVLTT(21), GRT(40),GZT(40)
      COMMON/L2/ RCOORD(35), ZCOORD(35), XP(10000), X1, Y1
      COMMON/L3/ SP(482), DS, STC, STH, STR, STZ, STR2, STZ2, STRZ2,
     1  STZR2, NSP, LSEG(5), LSEGS(5), SMC(4,9)
      COMMON/L4/ SPT(2), DUMP(8,60), NPRT3, CYCLE, INDSMP, LERROR, ISUR
      REAL NU
      INTEGER A1, A9, F, PC
      INTEGER CYCLE, TYPE, BND, EMP, FUL, LPB, OB, SUR, HYB, HSUR
      DATA BND,FUL,SUR,EMP,OB, HYB,HSUR,LPB/ 1,2,3,4,5,6,7,10000/
    2 FORMAT(1H1, 12A6)
    3 FORMAT(4F4.1,8F8.3)
    4 FORMAT(4F10.3,4I10)
    5 FORMAT('     IBAR='I3,'   JBAR='I3,'   DR='F8.3,'    DZ='F8.3,'    DT='
     1 F8.5,'    IPHM='I3,'   PC=' I3 ,'   ALP='F8.5)
    6 FORMAT('    BCB-R-T-L='4(F5.1,1X),1X,'A='F8.3,4X,'B='F8.3,6X,'C='
```

```
  1  F8.3/'      NU=' F9.3,4X,'EPS='F8.5,5X,'GR='E9.3,'     GZ='E9.3)
 7 FORMAT('       T=' F8.3, '  TWPLT=' F8.3, '   TWPRT=' F8.3,
  1 '   TWFIN='F8.3, '  LPR=' I2,                  '    TYPE=' I2)
 8 FORMAT(2I5,6F8.3)
 9 FORMAT('       NX='I3,'   NY='I3,'   XC='F8.3,5X,'YC='F8.3,'    XD='
  1 F8.3,5X,'YD='F8.3,'   U0='F8.3,'   V0='F8.3)
10 FORMAT('0     NX=0'10X,I5,' PARTICLES IN SYSTEM')
11 FORMAT('0 PARTICLE STORAGE EXHAUSTED')
12 FORMAT(1H1,12A6,  4X, ' T=' 1PE12.5, ' CYCLE=' I5,'   GR='1PE10.2,
  1  ' NPI=' I5)
13 FORMAT(6X,'I'4X,'J'6X,'F(I,J)'8X,'U(I,J)'8X,'V(I,J)'7X,'PSI(I,J)'
  1 5X,'THETA(I,J)'6X,'D(I,J)')
14 FORMAT(3X,2(2X,I3),I9, 5X, 5(2X,1PE12.5))
15 FORMAT(16I5)
16 FORMAT(/// ' NSGMTS =' I3, 5X, ' NJC1 =' I3, 5X, ' NJC2 =' I3, 5X,
  1  ' LQUDHT =' I3, 5X, 'NPRT2 ='I2,5X,'DEPS =' F5.2,5X, ' VEPS ='
  2  F5.2, 5X, ' DBETA =' F5.2// ' ISUR =' I2, 5X, ' STH =' F6.2, 5X,
  3  ' STR =' F6.2, 5X, ' STZ =' F6.2, 5X, ' DS =' F6.3, 5X,
  4  ' ICYCLE =' I3, 4X, ' IPLOT =' I2, 4X, ' COFST =' F9.6//
  5  ' RHO =' F10.4, 5X, ' THCKNS =' F5.2 , 5X, ' NGLVL =' I3///
  6  ' VALUES OF RCOORD(I), ZCOORD(I), GLVLTT(I), GRT(I), GZT(I), JHY
  7B(I), LHYB(I) AND NHYB(I) ')
17 FORMAT(// (16I8))
18 FORMAT( 8F10.3)
19 FORMAT(' T='1PE12.5,' CYCLE='I5,' ITER='I7)
20 FORMAT(10I5,2F8.3)
21 FORMAT('0 CYLINDRICAL COORDS ALLOWS NO INFLOW')
22 FORMAT('0 NO OBST OR I/O BOUNDARIES')
23 FORMAT('     TYPE='I2,'   L1='I5,'   L2='I5,'   L3='I5,'   L4='I5,
  1 '   L5='I5,'   L6='I5,'   L7='I5,'   NXB='I5,'   NYB='I5,'   UL='F8.3,
  2 '   UR='F8.3)
24 FORMAT( // (16F8.3))
25 FORMAT(/// ' **** DIMENSION CHECK **** ' 5X, I4, F10.5)
   READ 2, NAME
   READ 3, BCB,BCR,BCT,BCL,A,B,C,NU,EPS,GR,GZ,VSCALE
   READ  4, T, TWPLT, TWPRT, TWFIN, LPR, NPRT5
   READ 20, TYPE,L1,L2,L3,L4,L5,L6,L7,NXB,NYB,UL,UR
   IF(PC .EQ. 0) BCL=1.0
   PRINT 2, NAME
   PRINT 5, IBAR,JBAR,DR,DZ,DT,IPHM,PC,ALP
   PRINT 6, BCB,BCR,BCT,BCL,A,B,C,NU,EPS,GR,GZ
   PRINT 7, T, TWPLT, TWPRT, TWFIN,LPR, TYPE
   IF(TYPE .NE. 2) GO TO 27
   READ 15, NSGMTS, NJC1, NJC2, LQUDHT, NPRT2, ISUR, ICYCLE, IPLOT,
  1  NGLVL
   NGLVL1 = 2*NGLVL
   NGLVL2 = NGLVL + 1
   INDG = 1
   NJCELL = NJC1 + NJC2
   READ  18, (RCOORD(I),I=1,NSGMTS)
   READ  18, (ZCOORD(I),I=1,NSGMTS)
   READ  18, DEPS, VEPS, DBETA, SIGNVN, STH, STR, STZ, DS, COFST,
  1  RHO, THCKNS
   READ  18, (GLVLTT(I),I=1,NGLVL2)
   READ  18, (GRT(I),I=1,NGLVL1)
   READ  18, (GZT(I),I=1,NGLVL1)
   READ  15, (JHYB(I),I=1,NJCELL)
   READ  15, (LHYB(I),I=1,NJCELL)
```

```
      READ  15, (NHYB(I),I=1,NJCELL)
      PRINT 16, NSGMTS, NJC1, NJC2, LOUDHT, NPRT2, DEPS, VEPS, DBETA,
     1  ISUR, STH, STR, STZ, DS, ICYCLE, IPLOT, COEST, RHO, THCKNS,NGLVL
      PRINT 24, (RCOORD(I),I=1,NSGMTS)
      PRINT 24, (ZCOORD(I),I=1,NSGMTS)
      PRINT 26, (GLVLTT(I),I=1,NGLVL2)
      PRINT 26, (GRT(I),I=1,NGLVL1)
      PRINT 26, (GZT(I),I=1,NGLVL1)
   26 FORMAT(// (10F12.4))
      ICASE = 3
      PRINT 17, (JHYB(I),I=1,NJCELL)
      PRINT 17, (LHYB(I),I=1,NJCELL)
      PRINT 17, (NHYB(I),I=1,NJCELL)
      DBETA = DBETA/DZ
      LERROR = 0
   27 CONTINUE
      X=PC
      Y = DR*FLOAT(1-PC)
      DO 29 I=1,IP2
      RIP(I)=X+Y
      RRP(I)=1./RIP(I)
      R(I)=RIP(I)-.5*Y
      RRI(I)=1./R(I)
      Z=4.*R(I)
      RMORP(I)=(Z-Y)/(Z+Y)
      RPORM(I)=1./RMORP(I)
   29 X=X+Y
      IP1=IBAR+1
      JP1=JBAR+1
      IPHM=IPHM+1
      RDR=1./DR
      RDR2=RDR*RDR
      RDZ=1./DZ
      RDZ2=RDZ*RDZ
      DRODZ=DR*RDZ
      DZODR=DZ*RDR
      RDRDZ=RDR*RDZ
      COF1=2.*NU*RDR
      COF2=2.*NU*RDZ
      W=(1.+ALP)/(2.*(RDR2+RDZ2))
      DTODR=DT*RDR
      DTODZ=DT*RDZ
      TPLT = TWPLT
      TPRT = TWPRT
      TWPLT = TPLT + T
      TWPRT = TPRT + T
      CYCLE=0
      T = T + 0.000001
C TO SET UP THE MARGINES OF THE GEOMETRY OF THE PROB
      XR=IBAR*DR
      YT=JBAR*DZ
      XF = IBAR
      YF = JBAR
      IRPPC = 1023/AMAX0(IBAR,JBAR)
      N1 = 2*IBAR
      IF(PC .EQ. 0) IRPPC=1023/AMAX0(N1,JBAR)
      RPPC1 = FLOAT(IRPPC)
      IRASTR = IBAR*IRPPC
```

```
      JRASTR = JBAR*IRDRC
      IXL = (1023 - IRASTR)/2
      IXR = IXL + IRASTR
      IYB = (1023 - JRASTR)/2
      IYT = IYB + JRASTR
      DO 30 J=1,JP2
      DO 30 I=1,IP2
      F(I,J) = 4
      U(I,J) = 0.0
      V(I,J) = 0.0
      UTIL(I,J) = 0.0
      VTIL(I,J) = 0.0
      PSI(I,J) = 0.0
      THETA(I,J) = 0.0
   30 D(I,J) = 0.0
C  TO SPECIFY BDRY CELLS AND HYB CELLS
      DO 31 J=1,JP2
      F(1,J)= 1
   31 F(IP2,J) = 1
      IF(TYPE .EQ. 2) GO TO 33
      DO 32 I=2,IP1
      F(I,1) = 1
   32 F(I,JP2) = 1
   33 CONTINUE
      IF(TYPE .NE. 2) GO TO 37
      IEMP(1) = IBAR
      JEMP(1) = 1
      N1 = 2
      DO 35 N=1,NJCELL
      N2 = IP1 - LHYB(N) - NHYB(N)
      IF(N2 .LT. 1) GO TO 35
      IEMP(N1) = N2
      JEMP(N1) = JHYB(N) + 1
      N1 = N1 + 1
   35 CONTINUE
      IEMP(N1)= IBAR
      JEMP(N1) = JP2
      NREMPC = N1
      INDSMP = 0
      IP3 = IBAR + 3
      IH = IP2/2
      IH1 = IH + 1
      NSGMTH = NSGMTS/2
      JH = JP2/2
      COE3 = 6.2832*DR
      IF(PC .EQ. 1) COE3=THCKNS
      YH = JBAR/2
      XH = IBAR/2
      XW = IBAR
      IF(PC .EQ. 0) XH=0.0
      DO 36 N=1,NJCELL
      J = JHYB(N) + 1
      N1 = LHYB(N) + 1
      N2 = LHYB(N) + NHYB(N)
      F(IP2,J) = EMP
      IF(PC .EQ. 1) F(1,J)=EMP
      DO 36 I=N1,N2
      IF(PC .EQ. 1) F(IP3-I,J)=HYB
```

```
   36 F(I,J) = HYR
      STR2 = STR**2
      STZ2 = STZ**2
      STRZ2 = STR2/STZ2
      STZR2 = STZ2/STR2
      STC = STH + STZ
   37 CONTINUE
      X = 0.33*SQRT(ABS(GR)*XR)
      Y = 0.33*SQRT(ABS(GZ)*YT)
      DROU = AMAX1(X,Y,UL,UR)
      DO 38 J=1,JP2
      BCLT(J)=BCL
      ROS(J)=1.0
   38 BCRT(J)=BCR
      K=1
      NP=0
      IF(TYPE.EQ.1) GO TO 100
      PRINT 22
C TO ASSIGN MARKER PARTICLES AND INITIALIZE U(I,J) AND V(I,J)
   40 READ 8, NX,NY,XC,YC,XD,YD,U0,V0
      IF(NY.EQ.0) GO TO 80
      PRINT 9, NX, NY, XC, YC, XD, YD, U0, V0
      XTF=1./NX
      YTF=1./NY
      Y=YTF*.5
      DROU = AMAX1(DROU,ABS(U0),ABS(V0))
      IF(TYPE .EQ. 2) GO TO 141
      XC=XC*RDR
      YC=YC*RDZ
      XD=XD*RDR
      YD=YD*RDZ
   50 X=XTF*.5
   55 CONTINUE
      IF(X.GE.XC .AND.X.LE.XD .AND.Y.GE.YC .
     * AND.Y.LE.YD )GO TO 60
      GO TO 70
   60 I=X+2.
      J=Y+2.
      IF(F(I,J).EQ.OB)GO TO 70
      XP(K)=X
      XP(K+1)=Y
      K=K+2
      IF(K .GT. LPB) GO TO 75
      NP=NP+1
      F(I,J)=FUL
      IF(F(I+1,J).EQ.BND) U(I,J)=U(I,J)*ROS(J)+U0*(1.-ROS(J))
      IF(F(I+1,J).NE.BND.AND.F(I+1,J).NE.OB) U(I,J)=U0
      IF(F(I-1,J).NE.BND.AND.F(I-1,J).NE.OB) U(I-1,J)=U0
      IF(F(I,J-1).NE.BND.AND.F(I,J-1).NE.OB) V(I,J-1)=V0
      IF(F(I,J+1).NE.BND) V(I,J)=V0
   70 X=X+XTF
      IF(X.LT.IBAR)GO TO 55
      Y=Y+YTF
      IF(Y.LT.JBAR) GO TO 50
      GO TO 40
   75 PRINT 11
      RETURN
   80 IF(DROU .GE. 0.000001) DROU=VSCALE*DR/DROU
```

```
      PRINT 10, NP
      ASSIGN 250 TO KPET
      GO TO 645
C TO COMPUTE THE RASTER POINT COUNTS ASSOCIATED WITH THE GEOMETRY OF AN
C    INFLOW-OUTFLOW AND/OR OB PROB, AND TO DEFINE OB CELLS
  100 IF(PC .EQ. 0      .AND. UL .GE. 0.001) GO TO 120
      PRINT 23, TYPE, L1, L2, L3, L4, L5, L6, L7, NXB, NYB, UL, UR
      IF(L7.EQ.0) GO TO 105
      IL=L5+2
      IR=L6+1
      JTF=L7+1
      DO 104 J=2,JTF
      DO 104 I=IL,IR
  104 F(I,J)=OB
  105 IL1 = L1*IRPPC + IYB
      IL2 = L2*IRPPC + IYB
      IL3 = L3*IRPPC + IYB
      IL4 = L4*IRPPC + IYB
      IL5 = L5*IRPPC + IXL
      IL6 = L6*IRPPC + IXL
      IL7 = L7*IRPPC + IYB
      XDIS = 0.5/AMAX0(NXB,1)
      YDIS = 1.0/AMAX0(NYB,1)
      YFIR=(.5*YDIS)+L1
      NIN=NYB*(L2-L1)
      COLS=0.0
      DO 110 J=1,JP2
      IF(J.GE.(L1+2).AND.J.LE.(L2+1)) GO TO 107
  106 IF(J.GE.(L3+2).AND.J.LE.(L4+1)) GO TO 109
      GO TO 110
  107 U(1,J)=UL
      BCLT(J)=-1.0
      GO TO 106
  109 U(IP1,J)=UR
      BCRT(J)=-1.
      IF(UR .GE. 0.001) GO TO 110
      POS(J)=0.
      BCRT(J)=1.
  110 CONTINUE
      GO TO 40
  120 PRINT 21
      RETURN
C COMPUTATION OF RASTER PT CNTS, INITIALIZATION OF VEL COMP AND ASGMT OF
C    MARKER PARTICLES FOR FLDS IN AN AXISYMM TANK WITH ELLIPSOIDAL BKHDS
  141 CONTINUE
      DO 161 N=1,NSGMTS
      ISEGPR(N) = RPPCLL*RCOORD(N)
      ISEGPZ(N) = RPPCLL*ZCOORD(N)
      ISEGPR(N) = ISEGPR(N) + IXL
  161 ISEGPZ(N) = ISEGPZ(N) + IYB
      ISEGPR(NSGMTS+1) = ISEGPR(1)
      ISEGPZ(NSGMTS+1) = ISEGPZ(1)
      N1 = 0
      N2 = 1
      HTLQUD = LQUDHT
      NSGMT1 = NSGMTS - 1
  171 N1 = N1 + 1
      N2 = N2 + 1
```

```
      ASSIGN 173 TO KRET
      Z = RCOORD(N2) - RCOORD(N1)
      UL = RCOORD(N2+1) - RCOORD(N2)
      IF(ABS(Z) .GT. 0.0001) GO TO 172
      ASSIGN 174 TO KRET
      XD = RCOORD(N1)
      GO TO KRET
  172 XC = (ZCOORD(N2) - ZCOORD(N1))/Z
      YC = ZCOORD(N1) - XC*RCOORD(N1)
      IF(ABS(UL) .LT. 0.0001) GO TO 173
      XDIS = (ZCOORD(N2+1) - ZCOORD(N2))/UL
      YDIS = ZCOORD(N2) - XDIS*RCOORD(N2)
  173 XD = (Y - YC)/XC
      Z = (Y - YDIS)/XDIS
      IF(Y .GT. ZCOORD(N2)) XD=AMAX1(XD,Z)
  174 CONTINUE
      X = 0.5*XTE
      IF(ISUR .GT. 0 .AND. Y .GT. STH) CALL FCTN2(X,XTE,Y)
      X = X + XH
      IF(X .GT. XD) GO TO 177
      J = 2.0 + Y
  175 I = 2.0 + X
      XP(K) = X
      XP(K+1) = Y
      K = K + 2
      IF(PC .EQ. 0) GO TO 178
      XP(K) = XW - XP(K-2)
      XP(K+1) = Y
      K = K + 2
      NP = NP + 1
  178 IF(K .GT. LPB) GO TO 75
      NP = NP + 1
      IF(F(I,J) .EQ. HSUR .OR. F(I,J) .EQ. FUL) GO TO 177
      V(I,J) = VO
      IF(F(I+1,J) .NE. BND) U(I,J)=UO
      IF(F(I,J) .EQ. HYB) GO TO 176
      F(I,J) = FUL
      GO TO 177
  176 F(I,J) = HSUR
  177 X = X + XTE
      IF(X .LT. XD) GO TO 175
      Y = Y + YTE
      IF(Y .GE. HTLQUD) GO TO 179
      IF(Y .LT. ZCOORD(N2)) GO TO KRET
      GO TO 171
  179 IF(PC .EQ. 0) GO TO 181
      NSGMT1 = NSGMTH
      N3 = IP2 + 1
      DO 180 J=1,JP2
      F(IH,J) = F(IH+1,J)
      IF(F(IH,J) .EQ. HSUR .OR. F(IH,J) .EQ. FUL) U(IH,J)=UO
      DO 180 I=IH1,IP1
      U(IP2-I,J) = U(I,J)
      V(IP3-I,J) = V(I,J)
  180 F(IP3-I,J) = F(I,J)
  181 N1 = 0
      N2 = 1
      N3 = 1
```

```
      SIGN = 1.0
      N6 = 0
  182 N6 = N6 + 1
  183 SIGN = -1.0*SIGN
      NBHALF = N3 - 1
      IF(N6 .GE. 2) GO TO 186
      XBDRY(N3) = RCOORD(N2)
      YBDRY(N3) = ZCOORD(N2)
      N3 = N3 + 1
      N5 = NSGMTH + 1
  185 N1 = N1 + 1
      N2 = N2 + 1
      IF(N2 .GT. NSGMTS) GO TO 197
      IF(RC .EQ. 1 .AND. N2 .GT. N5) GO TO 197
      Z = RCOORD(N2) - RCOORD(N1)
      IF(ABS(Z) .LT. 0.0001) GO TO 183
      IF(SIGN .LT. 0.5 .AND. Z .LT. 0.001) GO TO 182
  186 XC = (ZCOORD(N2) - ZCOORD(N1))/Z
      YC = ZCOORD(N1) - XC*RCOORD(N1)
      XD = RCOORD(N2) + 0.0001*SIGN
      N4 = RCOORD(N1) + 0.0001
      IF(SIGN .GT. 0.5) N4=RCOORD(N1)+0.9999
      Z = N4
  187 Z = Z - SIGN
      IF(SIGN) 190,189,189
  189 IF(Z .LT. XD) GO TO 193
      GO TO 191
  190 IF(Z .GT. XD) GO TO 193
  191 XBDRY(N3) = Z
      YBDRY(N3) = Z*XC + YC
      N3 = N3 + 1
      GO TO 187
  193 XBDRY(N3) = RCOORD(N2)
      YBDRY(N3) = ZCOORD(N2)
      N3 = N3 + 1
      GO TO 185
  197 NBDRYP = N3 - 2
      IF(N6 .LT. 2) NBDRYP=NBDRYP-1
       IF(RC .EQ. 0) GO TO 203
      NBDRYP = 2*NBDRYP
      N = N3 - 2
      J = N3 -1
      DO 201 I=1,N
      J = J - 1
      XBDRY(N3) = XW - XBDRY(J)
      YBDRY(N3) = YBDRY(J)
  201 N3 = N3 + 1
  203 CONTINUE
      N1 = 0
      DO 211 N=1,NBDRYP
      N1 = N1 + 1
      N2 = N1 + 1
      XC = XBDRY(N2) - XBDRY(N1)
       IF(ABS(XC) .GT. 0.0005) GO TO 207
      N1 = N1 + 1
      N2 = N1 + 1
      XC = XBDRY(N2) - XBDRY(N1)
  207 YC = YBDRY(N2) - YBDRY(N1)
```

```
      ALPHA(N) = ABS(YC/XC)
      ALPHA(N) = ATAN(ALPHA(N))
      DL(N) = SQRT((XC*DR)**2 + (YC*DZ)**2)
      IF(XC .GT. 0.0) GO TO 209
      XC = ABS(XC/YC)
      GAMMA(N) = ATAN(XC) + 3.1416
      IF(YC .LT. 0.0) GAMMA(N)=6.2832-ATAN(XC)
      GO TO 210
  209 XC = ABS(YC/XC)
      GAMMA(N) = ATAN(XC) + 1.5708
      IF(YC .LT. 0.0) GAMMA(N)=1.5708-ATAN(XC)
  210 XBDRY(N) = 0.5*(XBDRY(N1) + XBDRY(N2))
  211 YBDRY(N) = 0.5*(YBDRY(N1) + YBDRY(N2))
      IF(ICUR .LT. 1) GO TO 215
      CALL ASCSMP(XH)
      IF(PC .EQ. 0) GO TO 215
      K = 1
      L = 2*NSP + 1
      M = 2*NSP - 1
      X = IBAR
      DO 213 I=1,NSP
      SPT(K) = X - SP(M)
      SPT(K+1) = SP(M+1)
      SPT(L) = SP(K)
      SPT(L+1) = SP(K+1)
      K = K + 2
      L = L + 2
  213 M = M - 2
      NSP = 2*NSP
      K = 1
      DO 214 I=1,NSP
      SP(K) = SPT(K)
      SP(K+1) = SPT(K+1)
  214 K = K + 2
  215 CONTINUE
      IF(NPEMPC .GT. 20) PRINT 216
  216 FORMAT( /// ' **** ERROR -- NPEMPC EXCEEDS 20 IN DO-LOOP 35. ')
      PRINT 218, NPEMPC, (IEMP(I),I=1,NPEMPC)
  218 FORMAT( /// ' VALUES OF NPEMPC, IEMP(I) AND JEMP(I) ' 10X, 'NPEMPC
     1 =' I3// (16I8))
      PRINT 17, (JEMP(I),I=1,NPEMPC)
      NBDCLL = NBDRYP + JBAR - NJCELL
      IF(PC .EQ. 1) NBDCLL=NBDCLL+JBAR-NJCELL
      PRINT 219, (ALPHA(I),I=1,NBDRYP)
  219 FORMAT(// ' VALUES OF ALPHA(N) AND DL(N) ' // (16F8.3))
      PRINT 24, (DL(I),I=1,NBDRYP)
      PRINT 220, NBDRYP, (XBDRY(I),I=1,NBDRYP)
  220 FORMAT(/// ' NBDRYP =' I4// ' VALUES OF XBDRY(I), YBDRY(I) AND GAM
     1MA(I) ' // (16F8.3))
      PRINT 24, (YBDRY(I),I=1,NBDRYP)
      PRINT 24, (GAMMA(I),I=1,NBDRYP)
      PERIMT = FLOAT(JBAR - NJCELL)*DZ
      IF(PC .EQ. 1) PERIMT=2*PERIMT
      DO 222 I=1,NBDRYP
  222 PERIMT = PERIMT + DL(I)
      N1 = 1
      N2 = 3
      N3 = 1
```

```
      N4 = NBDCLL
      N6 = NJC1 + 2
      N7 = JP1 - NJC2
      IF(PC .EQ. 1) N2=IH1+1
      DB(1) = 0.5*DL(1)
      DB(NBDCLL) = PERIMT - DB(1)
      DO 226 J=2,JP1
      DO 226 I=N2,IP1
      K = 1
      IF(J .GT. N7) K=IP1+N2-1
      N9 = F(K,J)
      GO TO (226,224,224,224,226,223,223), N9
  223 N1 = N1 + 1
      N3 = N3 +1
      N4 = N4 - 1
      DB(N3) = DB(N3-1) + 0.5*(DL(N1-1) + DL(N1))
      IF(PC .EQ. 1) DB(N4)=PERIMT-DB(N3)
      GO TO 226
  224 IF(K .NE. IP1 .OR. J .LT. N6 .OR. J .GT. N7) GO TO 226
      N3 = N3 + 1
      N4 = N4 - 1
      DB(N3) = DB(N3-1) + DZ
      IF(PC .EQ. 1) DB(N4)=PERIMT-DB(N3)
  226 CONTINUE
      IF(PC .EQ. 0) GO TO 40
      N1 = N1 + 1
      N = NBDCLL/2
      DB(N) = DB(N-1) + 0.5*(DL(N1-1) + DL(N1))
      DB(N+1) = DB(N) + 0.5*(DL(N1) + DL(N1+1))
      CALL CLKOUT
      GO TO 40
C TO INDICATE FUL AND SUR CELLS
C COMPUTATION OF FORCES AND MOMENT -- MOD MAY BE NEEDED IF DR NE DZ.
C MOMENT IS COMPUTED ABOUT TANK GC.
  250 CONTINUE
      IF(T .GT. GLVLTT(INDG+1)) INDG=INDG+1
      N1 = 2*INDG
      TEMP1 = GLVLTT(INDG+1) - GLVLTT(INDG)
      GR = GRT(N1-1) + (T - GLVLTT(INDG))*(GRT(N1) - GRT(N1-1))/TEMP1
      GZ = GZT(N1-1) + (T - GLVLTT(INDG))*(GZT(N1) - GZT(N1-1))/TEMP1
      DO 255 J=1,JP2
      DO 255 I=1,IP2
      P(I,J) = 0.0
      Q(I,J) = 0.0
  255 KF(I,J) = 0
      IND = 1
      INDEX = 1
      ITER = 0
      ITERVN = 0
      NPT=0
      K=1
  257 I=XP(K)+2.
      J=XP(K+1)+2.
      KF(I,J)=K
      K=K+2
      NPT=NPT+1
      IF(NPT.LT.NP)GO TO 257
      IF(TYPE .NE. 2) GO TO 260
```

```
      DO 258 J=1,JP2
      IF(F(IP2,J) .EQ. BND) GO TO 258
      F(IP2,J) = EMP
      KF(IP2,J) = 0
      IF(PC .EQ. 0) GO TO 258
      F(1,J) = EMP
      KF(1,J) = 0
  258 CONTINUE
      INDEMP = INDEMP + 1
      DO 260 J=1,NPEMPC
      K = JEMP(J)
      N1 = IP2 - IEMP(J)
      DO 259 I=N1,IP1
      F(I,K) = EMP
      KF(I,K) = 0
      IF(PC .EQ. 0) GO TO 259
      F(IP3-I,K) = EMP
      KF(IP3-I,K) = 0
  259 CONTINUE
  260 CONTINUE
C TO MODIFY THE VEL COMP OF A NEWLY CREATED EMP OR HYB CELL
      DO 265 J=2,JP1
      DO 265 I=2,IP1
      N9 = F(I,J)
      GO TO (265,265,261,265,265,265,263), N9
  261 IF(KF(I,J).NE.0) GO TO 265
      F(I,J)=EMP
      GO TO 264
  263 IF(KF(I,J) .NE. 0) GO TO 265
      F(I,J) = HYB
  264 IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) U(I,J)=0.0
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) U(I-1,J)=0.0
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) V(I,J)=0.0
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) V(I,J-1)=0.0
  265 CONTINUE
C TO SEE IF A SUR-CELL SHOULD BECOME FUL,OR A FUL-CELL SHOULD BECOME SUR
      DO 270 J=2,JP1
      DO 270 I=2,IP1
      IF(KF(I,J) .EQ. 0 .OR. F(I,J) .GE. OB) GO TO 270
      N1 = 0
      IF(F(I+1,J).EQ.EMP.OR.F(I-1,J).EQ.EMP.OR.F(I,J+1).EQ.EMP.OR.
     1  F(I,J-1) .EQ. EMP) N1=1
      IF(F(I+1,J).EQ.HYB .OR. F(I-1,J).EQ.HYB .OR. F(I,J+1).EQ.HYB .OR.
     1  F(I,J-1).EQ.HYB) N1=1
      N9 = F(I,J)
      GO TO (270,267,268), N9
  267 IF(N1 .EQ. 1) F(I,J)=SUR
      GO TO 270
  268 IF(N1 .EQ. 0) F(I,J)=FUL
  270 CONTINUE
      ASSIGN 280 TO KRET
      IF(TYPE .NE. 2) GO TO 650
      IF(ISUR .LT. 2 .OR. LERROR .NE. 0) GO TO 650
      CALL CURVE
      IF(LERROR .NE. 0) GO TO 650
      NPT = 2*NSP - 1
      K1 = 0
      K2 = 0
```

```
      K3 = 1
      DO 276 K=1,NPT,2
      I = CP(K) + 2.0
      J = CP(K+1) +2.0
      IF(I.EQ.K1 .AND. J .EQ. K2) GO TO 276
      NO = F(I,J)
      GO TO (276,274,274,276,276,276,274), NO
  274 K1 = I
      K2 = J
      CALL FORCE(K,K1,K2,K3,K4,K5,STU,STV)
      IF(K4 .GT. 0) P(K4,J) = STU*COEST
      IF(K5 .GT. 0) Q(I,K5) = STV*COEST
  276 CONTINUE
      N = CYCLE
      IF(N.EQ.0 .OR. N.EQ. 2 .OR. N.EQ.40 .OR. N.EQ.60 .OR. N.EQ.80)
     1  GO TO 277
      GO TO 650
  277 PRINT 278, CYCLE
  278 FORMAT( // ' **** CHECK P(I,J) AND Q(I,J) ****'10X, 'CYCLE =' I4)
      DO 279 J=1,JP2
      N = J
  279 PRINT 1484, (P(I,N),I=1,IP2), (Q(I,N),I=1,IP2)
      GO TO 650
  280 IF(CYCLE.NE.0)GO TO 300
C INITIAL SURFACE PRESSURE
      DO 290 J=2,JP1
      DO 290 I=IPHM,IP1
      IF(F(I,J).EQ.SUR) THETA(I,J)=(A+B*COS(C*(J-1.5)*DZ))/DT
  290 CONTINUE
C COMPUTATION OF PSEUDOPRESSURE OF SUR, HSUR, FUL AND HFUL CELLS
  300 ICHECK = 320
      DO 320 J=2,JP1
      DO 320 I=2,IP1
      IF(INDEX .EQ. 2 .AND. F(I,J) .NE. HSUR) GO TO 320
      IF(INDEX .EQ. 2) GO TO 318
      IF(CYCLE .NE. 0 .AND. F(I,J) .NE. HSUR) THETA(I,J)=0.0
      IF(F(I,J).EQ.SUR)GO TO 301
      IF(F(I,J) .NE. HSUR) GO TO 315
  301 N=0
      IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) N=N+1
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) N=N+2
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) N=N+4
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) N=N+8
      IF(NPRT3.EQ.1 .AND. N.EQ.0 .AND. CYCLE.LT.3) PRINT 656, I,J,ICHECK
      IF(N .EQ. 0) GO TO 317
      GO TO (305,310,320,305,320,320,320,310,320,320,320,320,320,320,320
     1 ), N
  305 THETA(I,J)=THETA(I,J)+COF1*(U(I,J)-U(I-1,J))
      GO TO 320
  310 THETA(I,J)=THETA(I,J)+COF2*(V(I,J)-V(I,J-1))
      GO TO 320
  315 IF(F(I,J) .NE. FUL) GO TO 320
      IF(TYPE .EQ. 1 .AND. UR .LE. 0.001) GO TO 320
  317 THETA(I,J) = GZ*(J-1.5)*DZ + GR*(I-1.5)*DR
      GO TO 320
  318 DO 319 K=1,NBDRYP
      N1 = XBDRY(K) + 2.0
      N2 = YBDRY(K) + 2.0
```

```
      IF(I .EQ. N1 .AND. J .EQ. N2 .AND. IDUMP(K) .EQ. 2) THETA(I,J)=
     1  THETA(I,J)+SIGNVN*DRETA*VN(K)
  310 CONTINUE
  320 CONTINUE
      IF(INDEX .EQ. 2) GO TO 600
      ASSIGN 370 TO KRET
C TO COMPUTE THE JUST-OUTSIDE TGTL VEL COMP
  330 ICHECK = 344
      DO 344 J=2,JP1
      DO 344 I=2,IP1
      IF(F(I,J).EQ.SUR)GO TO 331
      IF(F(I,J) .NE. HSUR) GO TO 344
  331 N = 0
      IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) N=N+1
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) N=N+2
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) N=N+4
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) N=N+8
      IF(NPRT3.EQ.1 .AND. N.EQ.0 .AND. CYCLE.LT.3) PRINT 656, I,J,ICHECK
      IF(N .EQ. 0) GO TO 344
      GO TO (333,333,344,333,333,333,344,333,333,333,344,333,333,333,
     1  344), N
  333 IF(F(I,J+1) .EQ. FUL .OR. F(I,J+1) .EQ. SUR .OR. F(I,J+1) .EQ.
     1  HSUR) GO TO 334
      GO TO 337
  334 GO TO (335,337,344,336,335,337,344,337,335,337,344,336,335,337), N
  335 IF(F(I+1,J+1) .EQ. EMP .OR. F(I+1,J+1) .EQ. HYB)
     * V(I+1,J)=V(I,J)-DRODZ*(U(I,J+1)-U(I,J))
      IF(N .EQ. 5 .OR. N .EQ. 13) GO TO 336
      GO TO 344
  336 IF(F(I-1,J+1) .EQ. EMP .OR. F(I-1,J+1) .EQ. HYB)
     * V(I-1,J)=V(I,J)+DRODZ*(U(I-1,J+1)-U(I-1,J))
      IF(N .NE. 12) GO TO 344
  337 IF(F(I+1,J) .EQ. FUL .OR. F(I+1,J) .EQ. SUR .OR. F(I+1,J) .EQ.
     1  HSUR) GO TO 339
      GO TO 344
  339 GO TO (344,341,344,344,344,341,344,343,344,341,344,343,344,341), N
  341 IF(F(I+1,J+1) .EQ. EMP .OR. F(I+1,J+1) .EQ. HYB)
     * U(I,J+1)=U(I,J)-DZODR*(V(I+1,J)-V(I,J))
      IF(N .EQ. 10 .OR. N .EQ. 14) GO TO 343
      GO TO 344
  343 IF(F(I+1,J-1) .EQ. EMP .OR. F(I+1,J-1) .EQ. HYB)
     * U(I,J-1)=U(I,J)+DZODR*(V(I+1,J-1)-V(I,J-1))
  344 CONTINUE
      IF(TYPE .NE. 2) GO TO 349
      N1 = NJC1 + 1
      V(IP2,N1) = V(IP1,N1)
      IF(PC .EQ. 1) V(1,N1)=V(2,N1)
      N1 = JP1 - NJC2
      V(IP2,N1) = V(IP1,N1)
      IF(PC .EQ. 1) V(1,N1)=V(2,N1)
C REFINEMENT IS NECESSARY IF PROB ON PROP RE-OR IS TO BE SMLTD.
      GO TO (345,346,349), ICASE
  345 V(14,17) = V(13,17)
      U(13,17) = -V(13,16)*COS(ALPHA(15))/SIN(ALPHA(15))
      IF(F(12,17) .EQ. EMP) U(12,17)=U(13,17)
      U(13,18) = -V(13,17)*COS(ALPHA(16))/SIN(ALPHA(16))
      IF(F(12,18) .EQ. EMP) U(12,18)=U(13,18)
      U(12,19) = -V(12,18)*COS(ALPHA(17))/SIN(ALPHA(17))
```

```
      IF(E(11,19) .EQ. HYD) U(11,19)=U(12,19)
      GO TO 349
  346 CONTINUE
  349 CONTINUE
      IF(TYPE .EQ.2) GO TO 352
      DO 350 I=2,IFAR
      U(I,1)=U(I,2)*RCO
  350 U(I,JP2)=U(I,JP1)*RCT
  352 CONTINUE
      DO 360 J=1,JP1
      IF(TYPE .EQ. 2 .AND. RC .EQ. 1 .AND. E(1,J) .EQ. EMP) GO TO 355
      V(1,J)=V(2,J)*RCLT(J)
  355 IF(TYPE .EQ. 2 .AND. E(IP2,J) .EQ. EMP) GO TO 360
      V(IP2,J) = V(IP1,J)*RCRT(J)
  360 CONTINUE
      GO TO KRET
  375 CONTINUE
  400 IF(CYCLE .EQ. 0) GO TO 415
      IF(TWPLT .LE. T) GO TO 405
  401 IF(TWPRT .LE. T) GO TO 407
      IF(NPRT2 .EQ. 1) PRINT 25, LHYD(29), ZCOORD(29)
  402 IF(TWFIN .LE. T) RETURN
  404 CYCLE = CYCLE + 1
      T = T + DT
      GO TO 500
  405 TWPLT = TWPLT + TPLT
  406 ASSIGN 401 TO KRET
      GO TO 420
  407 TWPRT = TWPRT + TPRT
  408 ASSIGN 402 TO KRET
      GO TO 430
  415 ASSIGN 416 TO KRET
      GO TO 420
  416 ASSIGN 404 TO KRET
      GO TO 430
  420 ASSIGN 424 TO KR1
C TO PLOT THE  GEOMETRY OF A PROB
  421 CALL FRAMEV(2)
      IF(TYPE .EQ. 1) GO TO 423
      IF(TYPE .EQ. 2) GO TO 460
      CALL LINEV(IXL,IYB,IXR,IYB)
      CALL LINEV(IXR,IYB,IXR,IYT)
      CALL LINEV(IXR,IYT,IXL,IYT)
      CALL LINEV(IXL,IYT,IXL,IYB)
  422 GO TO KR1
  423 CONTINUE
      CALL LINEV(IXL,IYB,IL5,IYB)
      CALL LINEV(IL5,IYB,IL5,IL7)
      CALL LINEV(IL5,IL7,IL6,IL7)
      CALL LINEV(IL6,IL7,IL6,IYB)
      CALL LINEV(IL6,IYB,IXR,IYB)
      CALL LINEV(IXR,IYB,IXR,IL3)
      CALL LINEV(IXR,IL4,IXR,IYT)
      CALL LINEV(IXR,IYT,IXL,IYT)
      CALL LINEV(IXL,IYT,IXL,IL2)
      CALL LINEV(IXL,IL1,IXL,IYB)
      GO TO 422
  424 NPT=0
```

```
      M = NC
      K = 1
      L = 0
      NPT = NP
  425 IX1 = XP(K)*RPPCLL
      IY1 = XP(K+1)*RPPCLL
      IX1 = IX1 + IXL
      IY1 = IY1 + IYB
      IF(TYPE .EQ. 2) GO TO 418
      L = 0
      IF(XP(K).LT.0.0 .OR. XP(K).GT.XF .OR. XP(K+1).LT.0.0 .OR.
     1   XP(K+1).GT.YF) L=1
      GO TO 419
  418 IF(CYCLE .GT. 0) CALL MARKER(CYCLE,K,L,NSGMT1,M,XW)
  419 CONTINUE
      IF(L .EQ. 0) CALL PLOTV(IX1,IY1,35,0)
      IF(L .EQ. 1) NPT=NPT-1
      K = K + 2
      NPT NPT+1
      IF(NPT.LT.NP) GO TO 425
      IF(LPP .EQ. 0) GO TO KRET
      ASSIGN 426 TO KR1
      IF(TYPE .EQ.2) GO TO 470
      GO TO 421
C FOR PLOTTING VEL VECTOR
  426 L = 0
      DO 428 J=2,JP1
      DO 428 I=2,IP1
      N9 = F(I,J)
      GO TO (428,427,427,428,428,428,427), N9
  427 IX1 = (FLOAT(I) - 1.5)*RPPCLL
      IY1 = (FLOAT(J) - 1.5)*RPPCLL
      IX2 = 0.5*(U(I-1,J) + U(I,J))*DROU*RPPCLL
      IY2 = 0.5*(V(I,J-1) + V(I,J))*DROU*RPPCLL
      IX1 = IX1 + IXL
      IY1 = IY1 + IYB
      IX2 = IX2 + IX1
      IY2 = IY2 + IY1
      IF(TYPE .NE. 2 .OR. F(I,J) .NE. HSUR) GO TO 429
      X1 = FLOAT(I) - 1.5
      Y1 = FLOAT(J) - 1.5
      CALL MARKER(CYCLE,0,L,NSGMT1,M,XW)
      IF(L .NE. 0) GO TO 428
  429 CALL LINEV (IX1,IY1,IX2,IY2)
      X = IX2 - IX1
      Y = IY2 - IY1
      N1 = SQRT(X**2 + Y**2)
      IF(N1 .GT. 7) CALL ARROW(IX1,IY1,IX2,IY2,6,2)
  428 CONTINUE
      GO TO KRET
  430 IF(LPP .EQ. 0) GO TO KRET
      PRINT 12, NAME,T,CYCLE, GR, NPI
      PRINT 13
C TO CHECK THE INCOMPRESSIBILITY OF A FLUID SYSTEM
  434 DO 440 JJ=1,JP2
      DO 440 I=1,IP2
      J=JP2+1-JJ
      D(I,J)=0.
```

```
      N9 = F(I,J)
      GO  TO (438,436,436,438,438,438,436), N9
  436 D(I,J) = RRI(I)*RDR*(RIP(I)*U(I,J) - RIP(I-1)*U(I-1,J))
     1  + RDZ*(V(I,J) - V(I,J-1))
  438 PRINT 14, I, J, F(I,J), U(I,J), V(I,J) , PSI(I,J), THETA(I,J),
     1  D(I,J)
  440 CONTINUE
      IF(TYPE .NE. 2) GO TO 448
      DO 230 I=1,NBDCLL
      FN(I) = 0.0
      FX(I) = 0.0
  230 FY(I) = 0.0
      DMOM = 0.0
      FTX = 0.0
      FTY = 0.0
      N1 = 0
      N2 = 2
      N3 = 0
      N4 = NBDCLL + 1
      N5 = NBDRY2 + 1
      N6 = NJC1 + 2
      N7 = JP1 - NJC2
      IF(PC .EQ. 1) N2=IH1
      DO 242 J=2,JP1
      DO 242 I=N2,IP1
      K = I
      IF(J .GT. N7) K=IP1+N2-I
      N9 = F(K,J)
      GO TO (242,236,236,236,242,232,232), N9
  232 N1 = N1 + 1
      N3 = N3 + 1
      N4 = N4 - 1
      N5 = N5 - 1
      Y = DZ*(YH - YBDRY(N1))
      X = DR*(XBDRY(N1) - XH)
      N=0
      IF(F(K+1,J) .EQ. EMP .OR. F(K+1,J) .EQ. HYB) N=N+1
      IF(F(K,J+1) .EQ. EMP .OR. F(K,J+1) .EQ. HYB) N=N+2
      IF(F(K-1,J) .EQ. EMP .OR. F(K-1,J) .EQ. HYB) N=N+4
      IF(F(K,J-1) .EQ. EMP .OR. F(K,J-1) .EQ. HYB) N=N+8
      FN(N3) = RHO*THETA(K,J)
      IF(N .EQ. 0) FN(N3)=FN(N3) + RHO*PSI(K,J)/DT
      FY(N3) = -COF3*FN(N3)*DL(N1)*COS(ALPHA(N1))
      IF(PC .EQ. 0) FY(N3)=XBDRY(N1)*FY(N3)
      IF(PC .EQ. 0) GO TO 234
      FN(N4) = RHO*THETA(IP3-K,J)
      FX(N4) = -COF3*FN(N4)*DL(N5)*SIN(ALPHA(N5))
      FY(N4) = -COF3*FN(N4)*DL(N5)*COS(ALPHA(N5))
      FX(N3) = COF3*FN(N3)*DL(N1)*SIN(ALPHA(N1))
      IF(J .GT. JH) FY(N4)=-FY(N4)
  234 IF(J .GT. JH) FY(N3)=-FY(N3)
      GO TO 240
  236 IF(K .NE. IP1 .OR. J .LT. N6 .OR. J .GT. N7) GO TO 242
      N3 = N3 + 1
      N4 = N4 - 1
      Y = DZ*(YH - FLOAT(J) + 1.5)
      FN(N3) = RHO*(THETA(K,J) + PSI(K,J)/DT)
      IF(PC .EQ. 0) GO TO 242
```

```
      FX(N3) = COF3*DZ*FN(N3)
      FN(N4) = RHO*(THETA(IP3-K,J) + PSI(IP3-K,J)/DT)
      FX(N4) =-COF3*DZ*FN(N4)
  240 FTY = FTY + FY(N3)
      IF(PC .EQ. 0) GO TO 242
      FTY = FTY + FY(N4)
      FTX = FTX + FX(N3) + FX(N4)
      RMOM = RMOM + X*(FY(N3) - FY(N4)) + Y*(FX(N3) + FX(N4))
  242 CONTINUE
      PRINT 441, RMOM, FTX, FTY
  441 FORMAT(// ' FORCES AND MOMENT EXERTED ON TK WALL DUE TO SL LIQUID'
     1  10X, 'RMOM =' 1PE12.4, 5X, 'FTX =' 1PE12.4, 5X, 'FTY =' 1PE12.4
     2  /// 5X, 'DB(N)' 11X, 'FN(N)' 11X, 'FX(N)' 11X, 'FY(N)' // )
      DO 442 N=1,NBDCLL
      I = N
  442 PRINT 444, DB(I), FN(I), FX(I), FY(I)
  444 FORMAT(8F16.5)
      PRINT 446, (LSEG(I),I=1,4), (LSEGS(I),I=1,5),
     1  ((SMC(I,K),K=1,9),I=1,4)
  446 FORMAT( /// ' **** VALUES OF LSEG(I), LSEGS(J) AND SMC(I,K) ' 10X,
     1  4I5, 10X, 5I5 // (9F14.4))
  448 CONTINUE
      IF(CYCLE .GT. 0 .OR. TYPE .NE. 2) GO TO KRET
      IF(NPRT2 .NE. 1) GO TO KRET
      ICHECK = 434
      PRINT 450, CYCLE, ICHECK
  450 FORMAT(/// ' **** CHECK CPU TIME FOR PLOTS AND COMPUTATIONS OF DO-
     1LOOPS 680 AND 344 ' 5X, ' CYCLE =' I2, '    ICHECK =' I4, '   ****')
      CALL CLKOUT
  452 FORMAT( // (14F9.4))
  454 FORMAT(14F9.4)
      DO 455 J=1,JP2
      N1 = J
      PRINT 452, (U(I,N1),I=1,IP2)
  455 PRINT 454, (V(I,N1),I=1,IP2)
      GO TO KRET
  460 DO 464 I=1,NSGMTS
      N = I + 1
      CALL LINEV(ISEGRR(I),ISEGRZ(I),ISEGRR(N),ISEGRZ(N))
  464 CONTINUE
      GO TO 422
  470 IF(IPLOT .NE. 1 .OR. ISUR .LT. 2) GO TO 421
      K = 1
      L = 0
      NPT = 0
  474 IX1 = SP(K)*RPPCLL
      IY1 = SP(K+1)*RPPCLL
      IX1 = IX1 + IXL
      IY1 = IY1 + IYP
      CALL PLOTV(IX1,IY1,43,0)
      K = K + 2
      NPT = NPT + 1
      IF(NPT .LT. NSP) GO TO 474
      GO TO 421
C  TO COMPUTE THE TILDE VEL COMP
  500 IF(TYPE .EQ. 2) GO TO 1410
  502 CONTINUE
      DO 520 J=2,JP1
```

```
      DO 520 I=2,IP1
      N9 = F(I,J)
      GO TO (520,504,504,520,520,520,504), N9
  504 IF(I .EQ. IP1) GO TO 510
      N1 = F(I+1,J)
      GO TO (510,506,506,510,510,510,506), N1
  506 ZIP=U(I,J)*RRP(I)*(R(I)*U(I-1,J)-R(I+1)*U(I+1,J))
      UVBR=0.25*(U(I,J)+U(I,J-1))*(V(I,J-1)+V(I+1,J-1))
      UVTR=0.25*(U(I,J)+U(I,J+1))*(V(I,J)   +V(I+1,J))
      UTIL(I,J)=U(I,J)+DT*(RDR*(ZIP+THETA(I,J)-THETA(I+1,J))
     * +RDZ*(UVBR-UVTR)+GR+NU*(RDZ2*(U(I,J+1)+U(I,J-1)-2.*U(I,J))
     * -RDRDZ*(V(I+1,J)-V(I+1,J-1)-V(I,J)+V(I,J-1)))+P(I,J))
  510 N1 = F(I,J+1)
      GO TO (520,512,512,520,520,520,512), N1
  512 ZIP=V(I,J)*(V(I,J-1)-V(I,J+1))
      UVTL=RIP(I-1)*0.25*(U(I-1,J)+U(I-1,J+1))*(V(I-1,J)+V(I,J))
      UVTR=RIP(I)   *0.25*(U(I,J)   +U(I,J+1))   *(V(I+1,J)+V(I,J))
      VTIL(I,J)=V(I,J)+DT*(RDZ*(ZIP+THETA(I,J)-THETA(I,J+1))+GZ
     * +RDR*RRI(I)*((UVTL-UVTR)-NU*(RIP(I)*(RDZ*(U(I,J+1)-U(I,J))
     * -RDR*(V(I+1,J)-V(I,J))) -RIP(I-1)   *(RDZ*(U(I-1,J+1)-U(I-1,J))
     * -RDR*(V(I,J)-V(I-1,J)))))+Q(I,J))
  520 CONTINUE
C TO COMPUTE D-TILDE OF FUL AND HFUL CELLS
      DO 525 J=2,JP1
  525 UTIL(IP1,J)=UTIL(IP1,J)*ROS(J)+UTIL(IP1-1,J)*(1.-ROS(J))
      ICHECK = 540
      DO 540 J=2,JP1
      DO 540 I=2,IP1
      N9 = F(I,J)
      GO TO (535,534,535,535,535,535,531), N9
  531 IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) GO TO 535
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) GO TO 535
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) GO TO 535
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) GO TO 535
      IF(NPRT3 .EQ. 1 .AND. CYCLE .LT. 3) PRINT 656, I, J, ICHECK
  534 D(I,J) = RRI(I)*RDR*(RIP(I)*UTIL(I,J) - RIP(I-1)*UTIL(I-1,J))
     1   + RDZ*(VTIL(I,J) - VTIL(I,J-1))
      GO TO 540
  535 PSI(I,J)=0.
  540 CONTINUE
C COMPUTATION OF PSI(I,J)
      IF(INDEX .GT. 1) GO TO 547
      ICHECK = 831
      IF(TYPE .EQ. 2) GO TO 810
  545 IF(INDEX .GT. 2) GO TO 642
      IF(INDEX .EQ. 2 .AND. ITER .GT. 0) GO TO 280
  547 ICHECK = 570
  550 IF(TYPE .EQ. 2) GO TO 557
      DO 555 I=2,IP1
      PSI(I,1)=PSI(I,2)
  555 PSI(I,JP2)=PSI(I,JP1)
  557 CONTINUE
      DO 560 J=2,JP1
      IF(F(1,J) .EQ. BND) PSI(1,J)=PSI(2,J)
  560 IF(F(IP2,J) .EQ. BND) PSI(IP2,J)=PSI(IP1,J)*ROS(J)
      IF(IND.EQ.0) GO TO 600
      IND=0
      ITER=ITER+1
```

```
      DO 570 J=2,JP1
      DO 570 I=2,IP1
      N9 = F(I,J)
      GO TO (570,563,570,570,570,570,562), N9
  562 IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) GO TO 570
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) GO TO 570
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) GO TO 570
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) GO TO 570
      IF(CYCLE .LT. 6 .AND. ITER .EQ. 1) PRINT 656, I, J, ICHECK
  563 PSIR=PSI(I+1,J)
      PSIB=PSI(I,J-1)
      PSIL=PSI(I-1,J)
      IF(TYPE.NE.1)GO TO 565
      IF(F(I+1,J).EQ.OB) PSIR=PSI(I,J)
      IF(F(I,J-1).EQ.OB) PSIB=PSI(I,J)
      IF(F(I-1,J).EQ.OB)PSIL=PSI(I,J)
  565 X=W*(-D(I,J)+RRI(I)*RDR2*(RIP(I)*PSIR+RIP(I-1)*PSIL)+RDZ2*(PSIB+
     1 PSI(I,J+1)))-ALP*PSI(I,J)
      Y=ABS(X)-ABS(PSI(I,J))
      Z=ABS(X)+ABS(PSI(I,J))
      PSI(I,J)=X
      IF(Z .LE. 0.000001) GO TO 570
      IF(ABS(Y/Z).GT.EPS)IND=1
  570 CONTINUE
      GO TO 550
C COMPUTATION OF U(I,J) AND V(I,J)
  600 CONTINUE
      DO 620 J=2,JP1
      DO 620 I=2,IP1
      N9 = F(I,J)
      GO TO (620,604,604,620,620,620,604), N9
  604 N1 = F(I+1,J)
      GO TO (610,606,606,610,610,610,606), N1
  606 U(I,J) = UTIL(I,J) - RDR*(PSI(I+1,J) - PSI(I,J))
  610 N1 = F(I,J+1)
      GO TO (620,612,612,620,620,620,612), N1
  612 V(I,J) = VTIL(I,J) - RDZ*(PSI(I,J+1) - PSI(I,J))
  620 CONTINUE
      ASSIGN 630 TO KRET
      N1 = ABS(VN(1))
      DO 621 N=2,NBDRYP
      N2 = ABS(VN(N))
      IF(N2 .GT. N1) N1=N2
  621 CONTINUE
      IF(ITER .LT. 2500 .AND. N1 .LT. 40) GO TO 625
      PRINT 622
  622 FORMAT( /// ' **** RUN IS STOPPED BECAUSE OF ITER OR VEL EXCEED TH
     1E GIVEN LIMITS (STMT 622). ')
      RETURN
  625 ITERVN = ITERVN + 1
      GO TO 650
  630 ASSIGN 640 TO KRET
      GO TO 330
  640 IF(TYPE .EQ. 2) GO TO 810
  642 ASSIGN 700 TO KRET
  645 IF(TYPE.NE.1)GO TO KRET
      DO 647 J=2,JP1
      DO 647 I=2,IP1
```

```
      IF(F(I,J).NE.OB)GO TO 647
      IF(F(I,J+1).NE.OB.AND.F(I+1,J).EQ.OB) U(I,J)=U(I,J+1)*BCB
      IF(F(I-1,J).NE.OB.AND.F(I,J+1).EQ.OB) V(I,J)=V(I-1,J)*BCB
      IF(F(I+1,J).NE.OB.AND.F(I,J+1).EQ.OB) V(I,J)=V(I+1,J)*BCB
  647 CONTINUE
      GO TO KRET
C TO COMPUTE THE VEL COM- OF A SUR OR HSUR CELL
  650 ICHECK = 680
      DO 680 J=2,J-1
      DO 680 I=2,IP1
      IF(F(I,J).EQ.SUR)GO TO 653
      IF(F(I,J) .NE. HSUR) GO TO 680
  653 N=0
      IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) N=N+1
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) N=N+2
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) N=N+4
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) N=N+8
      IF(NPRT3.EQ.1 .AND. N.EQ.0 .AND. CYCLE.LT.3) PRINT 656, I,J,ICHECK
  656 FORMAT( /  ' **** CELL F(I,J) IS TREATED AS A HFUL CELL, I =' I3,
     1  ' J =' I3, ' , IN DO-LOOP' I5, ',****')
      IF(N .EQ. 0) GO TO 680
      GO TO (661,662,663,664,661,665,662,667,668,659,658,669,661,662,
     1  661), N
  658 V(I,J-1) = 0.5*V(I-1,J-1)
      V(I,J) = 0.5*V(I-1,J)
  661 U(I,J)=(U(I-1,J)*RIP(I-1)-R(I)*DRODZ*(V(I,J)-V(I,J-1)))*RRP(I)
      GO TO 680
  659 V(I,J-1) = 0.5*(V(I-1,J-1) + V(I+1,J-1))
  662 V(I,J)=V(I,J-1)-DZODR*RRI(I)*(U(I,J)*RIP(I)-U(I-1,J)*RIP(I-1))
      GO TO 680
  663 U(I,J)=U(I-1,J)*RMORP(I)
      GO TO 666
  664 U(I-1,J)=(U(I,J)*RIP(I)+R(I)*DRODZ*(V(I,J)-V(I,J-1)))*RRP(I-1)
      GO TO 680
  665 U(I-1,J)=U(I,J)*RPORM(I)
  666 V(I,J)=V(I,J-1)-.25*DZ*(U(I,J)+U(I-1,J))*RRI(I)*(1 -PC)
      GO TO 680
  667 V(I,J-1)=V(I,J)+DZODR*RRI(I)*(U(I,J)*RIP(I)-U(I-1,J)*RIP(I-1))
      GO TO 680
  668 U(I,J)=U(I-1,J)*RMORP(I)
      GO TO 670
  669 U(I-1,J)=U(I,J)*RPORM(I)
  670 V(I,J-1)=V(I,J)+.25*DZ*RRI(I)*(U(I,J)+U(I-1,J))*(1 -PC)
  680 CONTINUE
      GO TO KRET
  700 NPT=0
      IF(TYPE .NE. 2) GO TO 709
      IF(ISUR .LT. 2) GO TO 709
      K = 1
      KN = 1
      NPN = 0
  701 IF(NPT .GE. NSP) GO TO 708
      ID = SP(K) + 2.0
      HPX = ID - 1.0 - SP(K)
      HMX = 1.0 - HPX
      JP = SP(K+1) + 1.5
      HPY = JP - 0.5 - SP(K+1)
      HMY = 1.0 - HPY
```

```
      UK = HPX*HMY*U(ID-1,JR+1) + HMX*HMY*U(ID,JR+1)
     1  + HPX*HPY*U(ID-1,JR) + HMX*HPY*U(ID,JR)
      IR = SP(K) + 1.5
      HPX = IR - 0.5 - SP(K)
      HMX = 1.0 - HPX
      JD = SP(K+1) + 2.0
      HPY = JD - 1.0 - SP(K+1)
      HMY = 1.0 - HPY
      VK = HPX*HMY*V(IR,JD) + HMX*HMY*V(IR+1,JD) + HPX*HPY*V(IR,JD-1)
     1  + HMX*HPY*V(IR+1,JD-1)
      SP(KN) = SP(K) + UK*DTODR
      SP(KN+1) = SP(K+1) + VK*DTODZ
      YC = SP(K+1)
      DO 702 N=1,NSGMT1
      N9 = N
      IF(YC .GT. ZCOORD(N) .AND. YC .LE. ZCOORD(N+1)) GO TO 704
  702 CONTINUE
      GO TO 707
  704 Z = RCOORD(N9+1) - RCOORD(N9)
      IF(ABS(Z) .LT. 0.001) GO TO 705
      XC = (ZCOORD(N9+1) - ZCOORD(N9))/Z
      YC = ZCOORD(N9) - XC*RCOORD(N9)
      XD = (SP(K+1) - YC)/XC
      GO TO 706
  705 XD = RCOORD(N9)
  706 IF(SP(K) .GT. XD) GO TO 707
      NPN = NPN + 1
      KN = KN + 2
  707 K = K + 2
      NPT = NPT + 1
      GO TO 701
  708 NSP = NPN
      IF(ISUR .EQ. 3 .AND. INDSMP .GE. ICYCLE) CALL CHKSMP
  709 NPT = 0
      NPN=0
      K=1
      KN=1
C TO COMPUTE THE MOVEMENTS OF A MARKER PARTICLE
  710 IF(NPT.GE.NP)GO TO 735
      ID=XP(K)+2.
      HPX=ID-1.-XP(K)
      HMX=1.-HPX
      JR=XP(K+1)+1.5
      HPY=JR-0.5-XP(K+1)
      HMY=1.-HPY
      UK=HPX*HMY*U(ID-1,JR+1)+HMX*HMY
     * *U(ID,JR+1)+HPX*HPY*U(ID-1,JR)+HMX*HPY*U(ID,JR)
      IR=XP(K)+1.5
      HPX=IR-0.5-XP(K)
      HMX=1.-HPX
      JD=XP(K+1)+2.
      HPY=JD-1.-XP(K+1)
      HMY=1.-HPY
      VK=HPX*HMY*V(IR,JD)+HMX*HMY*V(IR+1,JD)+HPX*HPY
     * *V(IR,JD-1)+HMX*HPY*V(IR+1,JD-1)
      XP(KN)=XP(K)+UK*DTODR
      XP(KN+1)=XP(K+1)+VK*DTODZ
      I=XP(KN)+2.
```

```
      J=XP(KN+1)+2.
      IF(TYPE.NE.1)GO TO 715
      IF(XP(KN).GE.IBAR)GO TO 720
  715 KN=KN+2
      NPN=NPN+1
  720 IF(F(I,J).EQ.EMP)GO TO 760
      IF(F(I,J) .EQ. HYB) GO TO 764
  730 K=K+2
      NPT=NPT+1
      GO TO 710
  735 NP=NPN
      IF(TYPE.NE.1) GO TO 250
  740 X=UL*RDR*(T+DT)-XDIS*(2.*COLS+1.)
      IF(X.LT.0.)GO TO 250
C TO INTRODUCE NEW MARKER PARTICLES IN CASE OF AN INFLOW-OUTFLOW PROB
      COLS=COLS+1.
      Y=YFIR
      NPN=NP+NIN
  750 XP(KN)=X
      XP(KN+1)=Y
      KN=KN+2
      I=X+2.
      J=Y+2
      NP=NP+1
      Y=Y+YDIS
      IF(F(I,J).NE.EMP)GO TO 755
      F(I,J)=SUR
      U(I,J)=UL
  755 IF(KN .GT. LPR) GO TO 75
      IF(NP.LT.NPN)GO TO 750
      GO TO 740
  760  F(I,J)=SUR
      GO TO 770
  764 F(I,J) = HSUR
  770 IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) U(I,J)=U(ID,JD)
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) U(I-1,J)=U(ID-1,JD)
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) V(I,J)=V(ID,JD)
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) V(I,J-1)=V(ID,JD-1)
      GO TO 730
  810 GO TO (815,881), INDEX
  815 N1 = 0
      DO 831 N=1,NBDRYP
      I = XBDRY(N) + 2.0
      J = YBDRY(N) + 2.0
      IDUMP(N) = 0
      VN(N) = 0.0
      IF(F(I,J) .NE. HSUR) GO TO 831
      IF(F(I+1,J) .EQ. EMP .OR. F(I+1,J) .EQ. HYB) GO TO 821
      IF(F(I,J+1) .EQ. EMP .OR. F(I,J+1) .EQ. HYB) GO TO 821
      IF(F(I-1,J) .EQ. EMP .OR. F(I-1,J) .EQ. HYB) GO TO 821
      IF(F(I,J-1) .EQ. EMP .OR. F(I,J-1) .EQ. HYB) GO TO 821
      IF(CYCLE .LT. 4) PRINT 656, I, J, ICHECK
      GO TO 831
  821 IF(N1 .EQ. 0) N1=N
      IDUMP(N) = 1
      N2 = N
  831 CONTINUE
      K = 1
```

```
      NPT   =1
      N4  =  1
      N9  =  0
      INDEX =  2
      DO 837 I=1,8
      DO 837 J=1,NBDRYP
  837 DUMP(I,J) = 0.0
  835 IF(NPT .GT. NP) GO TO 867
      I  =  XP(K)
      J  =  XP(K+1)
      DO 851 N=N1,N2
      IF(IDUMP(N) .EQ. 0 .AND. IDUMP(N-1) .EQ. 2) N4=N
      IF(IDUMP(N) .EQ. 0 .AND. IDUMP(N+1) .EQ. 2) N4=N
      IF(IDUMP(N) .NE. 1) GO TO 851
      N6  =  XBDRY(N)
      N7  =  YBDRY(N)
      IF(I .EQ. N6 .AND. J .EQ. N7) GO TO 841
      GO TO 851
  841 Z = (XP(K) - XBDRY(N))*COS(GAMMA(N)) + (XP(K+1) - YBDRY(N))*SIN(
     1  GAMMA(N))
      IF(ABS(Z) .GT. DEPS) GO TO 851
      IDUMP(N)  =  2
      N6  =  XBDRY(N)  +  2.0
      Z  =  N6
      DUMP(1,N)  =  Z  -  1.0  -  XBDRY(N)
      DUMP(2,N)  =  1.0  -  DUMP(1,N)
      N7  =  YBDRY(N)  +  1.5
      Z  =  N7
      DUMP(3,N)  =  Z  -  0.5  -  YBDRY(N)
      DUMP(4,N)  =  1.0  -  DUMP(3,N)
      N6  =  XBDRY(N)  +  1.5
      Z  =  N6
      DUMP(5,N)  =  Z  -  0.5  -  XBDRY(N)
      DUMP(6,N)  =  1.0  -  DUMP(5,N)
      N7  =  YBDRY(N)  +  2.0
      Z  =  N7
      DUMP(7,N)  =  Z  -  1.0  -  YBDRY(N)
      DUMP(8,N)  =  1.0  -  DUMP(7,N)
      N9  =  N9  +  1
      N4  =  K  +  1
      Z  =  (XP(K)  -  XBDRY(N))*COS(GAMMA(N))  +  (XP(K+1)  -  YBDRY(N))*SIN(
     1  GAMMA(N))
      IF(NPRT3.EQ.1 .AND. N9.LT.51) PRINT 844, N1,N2,N,XP(K),XP(N4),Z
  844 FORMAT( / ' CHECK DO-LOOP 851 ' 10X, 3I5, 3F12.5)
      N4  =  N
      GO TO 855
  851 CONTINUE
  855 IF(N4 .NE. N1) GO TO 859
      N1  =  N1  +  1
  859 IF(N4 .NE. N2) GO TO 863
      N2  =  N2  -  1
  863 IF(N2 .LT. N1) GO TO 871
      K  =  K  +  2
      NPT  =  NPT  +  1
      GO TO 835
  867 PRINT 868, CYCLE, NPT, (IDUMP(I),I=1,NBDRYP)
  868 FORMAT(/// '**** NOTE -- THERE IS A HSUR BUT Z IS LT DEPS. *******
     1* '// 2I8// (16I8))
```

```
871 IF(CYCLE .GT. 3 .OR. NPRT2 .NE. 1) GO TO 545
    PRINT 872, CYCLE, NPT, (IDUMP(I),I=1,NBDRYP)
872 FORMAT(/// ' **** VALUES OF CYCLE, NPT, IDUMP(I) AND COMPUTED AREA
   1S **** '// 2I8// (16I8))
    DO 875 I=1,8
    N1 = I
875 PRINT 24, (DUMP(N1,J),J=1,NBDRYP)
    GO TO 545
881 CONTINUE
    DO 891 N=1,NBDRYP
    IF(IDUMP(N) .NE. 2) GO TO 891
    ID = XBDRY(N) + 2.0
    JR = YBDRY(N) + 1.5
    IR = XBDRY(N) + 1.5
    JD = YBDRY(N) + 2.0
    UK = DUMP(1,N)*DUMP(4,N)*U(ID-1,JR+1) + DUMP(2,N)*DUMP(4,N)*U(ID,
   1 JR+1) + DUMP(1,N)*DUMP(3,N)*U(ID-1,JR) + DUMP(2,N)*DUMP(3,N)*
   2 U(ID,JR)
    VK = DUMP(5,N)*DUMP(8,N)*V(IR,JD) + DUMP(6,N)*DUMP(8,N)*V(IR+1,JD)
   1 + DUMP(5,N)*DUMP(7,N)*V(IR,JD-1) + DUMP(6,N)*DUMP(7,N)*V(IR+1,
   2 JD-1)
    VN(N) = UK*COS(GAMMA(N)) + VK*SIN(GAMMA(N))
891 CONTINUE
    N1 = 0
    DO 901 N=1,NBDRYP
901 IF(ABS(VN(N)).LT. VEPS) N1=N1+1
    IND = 1
    IF(N1 .EQ. NBDRYP) INDEX=3
    IF(NPRT2 .NE. 1) GO TO 545
    CALL CLKOUT
    PRINT 910, CYCLE, ITER, (VN(I),I=1,NBDRYP)
910 FORMAT(/// ' **** VALUES OF CYCLE, ITER AND VN(I) **** '// 2I8//
   1 (16F8.4))
    GO TO 545
1410 GO TO (1420,1440), INDEX
1420 DO 1430 J=1,JP2
    DO 1430 I=1,IP2
    G(I,J) = U(I,J)
1430 H(I,J) = V(I,J)
    GO TO 1460
1440 IF(CYCLE .GT. 2 .OR. NPRT2 .NE. 1 .OR. ITERVN .GT. 3) GO TO 1445
    CALL CLKOUT
    PRINT 1442
1442 FORMAT( // ' CHECK U(I,J), V(I,J) AND PSI(I,J) BEFORE ADJUSTING TH
   1ETA(I,J) ')
    DO 1443 J=1,JP2
    N1 = J
1443 PRINT 1484, (U(I,N1),I=1,14), (V(I,N1),I=1,14), (PSI(I,N1),I=1,14)
1445 CONTINUE
    DO 1450 J=1,JP2
    DO 1450 I=1,IP2
    U(I,J) = G(I,J)
1450 V(I,J) = H(I,J)
1460 IF(CYCLE .GT. 3 .OR. NPRT2 .NE. 1 .OR. ITERVN .GT. 3) GO TO 502
    PRINT 1470, CYCLE, ITER, INDEX, SIGNVN
1470 FORMAT( // ' CHECK CYCLE, ITER, INDEX, SIGNVN AND THETA(I,J)' 10X,
   1 3I6, F6.2)
    DO 1480 J=1,JP2
```

```
      N1 = J
 1480 PRINT 1484, (THETA(I,N1),I=1,14),(U(I,N1),I=1,14),(V(I,N1),I=1,14)
 1484 FORMAT( / (14F9.4))
      GO TO 502
      END
@FOR,IS S03,S03
      SUBROUTINE CLKOUT
      CALL SCLOCK(DATE,TIME,FSEC,F60SEC)
      WRITE(6,1000) TIME
 1000 FORMAT( 6H0TIME= A12)
C **** MODIFIED FOR EXEC 8 VERSION
      CALL CPUTIM(ITIM)
      FSEC = FLOAT(ITIM)/1.E6
      WRITE(6,2000) FSEC
 2000 FORMAT( 13H0FSEC (CPU) =  F14.4)
      RETURN
      END
@ASM,IL CPUTIM,CPUTIM          . USE AS    CALL CPUTIM(ITIM)
$(1)          AXR$             . WHERE ITIM IS ELAPSED CPU TIME
CPUTIM*       LA      A0,(23,ARRAY)    . IN MICROSECONDS
              ER      PCT$
              LA      A0,ARRAY+22
              MSI,XU  A0,200
              SA      A0,*0,X11
              J       2,X11
ARRAY         RES     23
              END
@FOR,IS S04,S04
      SUBROUTINE MARKER(CYCLE,K,L,NSGMT1,M,XW)
      COMMON/L2/ RCOORD(35), ZCOORD(35), XP(10000), X1, Y1
      L = 0
      YC = XP(K+1)
      IF(K .EQ. 0) YC=Y1
      DO 3020 N=1,NSGMT1
      N9 = N
      IF(YC .GT. ZCOORD(N) .AND. YC .LE. ZCOORD(N+1)) GO TO 3030
 3020 CONTINUE
      L = 1
      RETURN
 3030 Z = RCOORD(N9+1) - RCOORD(N9)
      IF(ABS(Z) .LT. 0.001) GO TO 3035
      XC = (ZCOORD(N9+1) - ZCOORD(N9))/Z
      YC = ZCOORD(N9) - XC*RCOORD(N9)
      XD = (XP(K+1) - YC)/XC
      IF(K .EQ. 0) XD=(Y1-YC)/XC
      GO TO 3040
 3035 XD = RCOORD(N9)
 3040 IF(K .NE. 0 .AND. XP(K) .GT. XD) L=1
      IF(K .EQ. 0 .AND. X1 .GT. XD) L=1
      IF(M .EQ. 0 .OR. L .EQ. 1) RETURN
      XD = XW - XD
      IF(K .NE. 0 .AND. XP(K) .LT. XD) L=1
      IF(K .EQ. 0 .AND. X1 .LT. XD) L=1
      RETURN
      END
@FOR,IS S05,S05
      SUBROUTINE ASGSMP(XH)
      COMMON/L3/ SP(482), DS, STC, STH, STR, STZ, STR2, STZ2, STRZ2,
```

```
     1  STZR2, NSP, LSEG(5), LSEGS(5), SMC(4,9)
      K = 3
      N = 1
      NSP = 1
      X1 = 0.1*DS + XH
      Y1 = STH
      XTF = DS
      YD = STC - DS
      YC = STH + 0.5*STZ
      SP(1) = X1
      SP(2) = Y1
 3110 NSP = NSP +1
      IF(NSP .GT. 241) GO TO 3160
 3120 GO TO (3130,3140), N
 3130 X2 = X1 + XTF
      Y2 = STC - SQRT(STZ2 - STZR2*(X2-XH)**2)
      Z = SQRT((X2 - X1)**2 + (Y2 - Y1)**2)
      IF(Z .LT. DS) GO TO 3150
      XTF = 0.98*XTF
      GO TO 3120
 3140 Y2 = Y1 + XTF
      X2 = SQRT(STR2 - STRZ2*(Y2 - STC)**2) + XH
      Z = SQRT((X2 - X1)**2 + (Y2 - Y1)**2)
      IF(Z .GT. DS) GO TO 3150
      XTF = 1.02*XTF
      GO TO 3120
 3150 SP(K) = X2
      SP(K+1) = Y2
      IF(Y2 .GT. YD) RETURN
      K = K + 2
      X1 = X2
      Y1 = Y2
      IF(N .EQ. 2 .OR. Y2 .LT. YC) GO TO 3110
      N = 2
      XTF = Y2 - SP(K-3)
      GO TO 3110
 3160 PRINT 3164
 3164 FORMAT( /// ' **** ERROR -- ASSIGNMENT OF SURFACE MARKER PARTICLES
     1 EXCEEDS 241 LIMIT. **** ')
      RETURN
C **** A SURFACE MARKER PARTICLE WILL BE ADDED OR ELIMINATED BETWEEN ***
C        TWO NEIGHBORING PARTICLES.
      ENTRY CHKSMP
      COMMON/L4/ SPT(2), DUMP(8,60), NPRT3, CYCLE, INDSMP, LERROR, ISUR
      INDSMP = 0
      X2 = 0.4*DS
      Y2 = 1.6*DS
      J = 3
      K = 3
      SPT(1) = SP(1)
      SPT(2) = SP(2)
      DO 3280 N=2,NSP
      IF(J .GT. 481) GO TO 3299
      X1 = SP(K) - SPT(J-2)
      Y1 = SP(K+1) - SPT(J-1)
      Z = SQRT(X1**2 + Y1**2)
      IF(Z .LT. X2) GO TO 3280
      IF(Z .GT. Y2) GO TO 3220
```

```
 3210 SPT(J) = SP(K)
      SPT(J+1) = SP(K+1)
      J = J + 2
      GO TO 3200
 3220 IF(ABS(X1) .GE. ABS(Y1)) GO TO 3225
      N1 = 1
      Y1 = SPT(J-1) + 0.5*Y1
      GO TO 3230
 3225 N1 = 2
      X1 = SPT(J-2) + 0.5*X1
C **** IN CASE OF N=NSP, CURVE FITTING IS MADE BY USING N-3, N-2,    ****
C           N-1 AND N INSTEAD OF N-2, N-1, N AND N+1.
 3230 L= K
      IF(N .EQ. NSP) L=L-2
      L1 = L - 4
      L2 = L - 2
      L3 = L + 2
      C1 = SP(L1) + SP(L2) + SP(L) + SP(L3)
      C2 = SP(L1)**2 + SP(L2)**2 + SP(L)**2 + SP(L3)**2
      C3 = SP(L1)**3 + SP(L2)**3 + SP(L)**3 + SP(L3)**3
      C4 = SP(L1)**4 + SP(L2)**4 + SP(L)**4 + SP(L3)**4
      C5 = SP(L-3) + SP(L-1) + SP(L+1) + SP(L+3)
      C6 = SP(L1)*SP(L-3) + SP(L2)*SP(L-1) + SP(L)*SP(L+1)
     1      + SP(L3)*SP(L+3)
      C7 = SP(L-3)*SP(L1)**2 + SP(L-1)*SP(L2)**2 + SP(L+1)*SP(L)**2
     1      + SP(L+3)*SP(L3)**2
      C8 = C2**2 - C1*C3
      C9 = C1*C5 - 4.0*C6
      C10 = C1*C2 - 4.0*C3
      C11 = C1**2 - 4.0*C2
      C12 = C2*C6 - C1*C7
      C13 = C2*C3 - C1*C4
      N2 = 3
      IF(ABS(C8) .LT. 0.00001) N2=1
      IF(ABS(C11) .LT. 0.00001) N2=N2-1
      GO TO (3240,3245,3250), N2
 3240 COF3 = C12/C13
      COF4 = (C9 - C10*COF3)/C11
      GO TO 3255
 3245 COF3 = C9/C10
      COF4 = (C12 - C13*COF3)/C8
      GO TO 3255
 3250 COF3 = (C8*C9 - C11*C12)/(C8*C10 - C11*C13)
      COF4 = (C9 - C10*COF3)/C11
 3255 COF5 = 0.25*(C5 - C1*COF4 - C2*COF3)
      GO TO (3260,3265), N1
 3260 COF6 = SQRT(1.0 - 4.0*COF3*(COF5 - Y1)/(COF4**2))
      X1 = -0.5*COF4*(1.0 - COF6)/COF3
      COF7 = COF5 + COF4*X1 + COF3*X1**2
      IF(ABS(Y1-COF7) .LT. 0.0001) GO TO 3270
      X1 = -0.5*COF4*(1.0 + COF6)/COF3
      COF7 = COF5 + COF4*X1 + COF3*X1**2
      IF(ABS(Y1-COF7) .LT. 0.0001) GO TO 3270
      PRINT 3262, CYCLE, K
 3262 FORMAT( /// ' **** ERROR - COMPUTATION OF X1 FROM Y1 IS INCORRECT.
     1  ' 10X, ' CYCLE =' I5, 5X, ' K =' I4)
      RETURN
 3265 Y1 = COF5 + COF4*X1 + COF3*X1**2
```

```
3270 SPT(J) = X1
     SPT(J+1) = Y1
     J = J + 2
     IF(J .GT. 481) GO TO 3299
     GO TO 3210
3280 K = K + 2
     N2 = J + 1
     IF(NOPT3 .NE. 1) GO TO 3290
     N1 = 2*NSP
     PRINT 3284, CYCLE,NSP, J
3284 FORMAT(///'***** VALUES OF CYCLE, NSP, NPT, SP(I) AND SPT(I)' 3I10)
     PRINT 3286, (SP(I),I=1,N1)
3286 FORMAT( // (16F8.2))
     PRINT 3286, (SPT(I),I=1,N2)
3290 NSP = (J+1)/2
     DO 3295 I=1,N2
3295 SP(I) = SPT(I)
     RETURN
3299 PRINT 7164
     RETURN
     ENTRY CURVE
C **** THE FS OF A FLUID IS PRESENTLY RESTRICTED TO A SIMPLE CONTOUR
C  DEFINED BY NOT MORE THAN 4 FCTNS WHICH ARE MONOTONICALLY IN R OR Z.
C  NO CURVE CAN HAVE 61 OR MORE PTS AND ASSIGNMENT OF FS MARKERS FOLLOWS
C  THE CCW PATH OF AN EMPTY DOMAIN.
     DIMENSION SMX(66), SMY(66), C(9)
     XC = SP(9) - SP(1)
     YC = SP(10)- SP(2)
     IF(XC .GT. 0 .AND. ABS(YC) .LT. XC) GO TO 3410
     IF(XC .LT. 0 .AND. ABS(YC) .LT. ABS(XC)) GO TO 3415
     IF(YC .GT. 0) GO TO 3420
     LSEG(1) = 4
     GO TO 3425
3410 LSEG(1) = 1
     GO TO 3425
3415 LSEG(1) = 3
     GO TO 3425
3420 LSEG(1) = 2
3425 LSEGS(1) = 1
     N1 = 2*(NSP - 4) - 1
     N2 = LSEG(1)
     N3 = 1
     N5 = 1
     DO 3490 K=3,N1,2
     N3 = N3 + 1
     XC = SP(K) - SP(K-2)
     YC = SP(K+1) - SP(K-1)
     GO TO (3440,3450,3460,3470), N2
3440 Z = YC/XC
     IF(N3 .GT. 61) GO TO 3447
     IF(Z .LT. 1.2) GO TO 3490
     CALL FCTN1(K,1,2,L,1)
     IF(L .EQ. 0) GO TO 3490
3447 LSEG(N5+1) = 2
     GO TO 3480
3450 Z = XC/YC
     IF(N3 .GT. 61) GO TO 3457
     IF(Z .GT. -1.2) GO TO 3490
```

```
      CALL FCTN1(K,-1.2,L,2)
      IF(L .EQ. 0) GO TO 3490
 3457 LSEG(NS+1) =3
      GO TO 3480
 3460 Z = YC/XC
      IF(N3 .GT. 61) GO TO 3467
      IF(Z .LT. 1.2) GO TO 3490
      CALL FCTN1(K,1.2,L,1)
      IF(L .EQ. 0) GO TO 3490
 3467 LSEG(NS+1) = 4
      GO TO 3480
 3470 Z = YC/XC
      IF(N3 .GT. 61) GO TO 3477
      IF(Z .GT. -1.2) GO TO 3490
      CALL FCTN1(K,-1.2,L,2)
      IF(L .EQ. 0) GO TO 3490
 3477 LSEG(NS+1) = 1
 3480 NS = NS + 1
      IF(NS .GT. 5) GO TO 3495
      IF(K .EQ. N1) GO TO 3494
      N3 = LSEG(NS)
      LSEGS(NS) = K
      N7 = 1
 3490 CONTINUE
      LSEGS(NS+1) = 2*NSP - 1
      GO TO 3510
 3494 LSEG(NS) = 0
      LSEGS(NS) = 2*NSP - 1
      GO TO 3510
 3495 N1 = CYCLE - 1
      PRINT 3496, N1
 3496 FORMAT( ///  '***** ERROR -- PRESENTLY, FREE SURFACE IS LIMITED TO
     1 4 SEGMENTS.  SURFACE TENSION EFFECT IS NEGLECTED AFTER CYCLE' I4,
     2  ' (STMT 3495).')
      LERROR = 1
      RETURN
 3510 DO 3550 L=1,4
      IF(LSEG(L) .EQ. 0) GO TO 3550
      N1 = LSEGS(L)
      IF(L .GT. 1) N1=N1-10
      N2 = LSEGS(L+1) - 2
      N = (N2 - N1)/2
      DO 3520 I=N1,N2,2
      J = (I + 1)/2
      SMX(J) = SP(I)
 3520 SMY(J) = SP(I+1)
      N3 = LSEG(L)
      GO TO (3525,3530,3525,3530),N3
 3525 CALL LSQPF1(SMX,SMY,0,N,7,C,IERR)
      GO TO 3540
 3530 DO 3535 I=N1,N2,2
 3535 SMX(I) = - SMX(I)
      CALL LSQPF1(SMY,SMX,0,N,7,C,IERR)
 3540 IF(IERR .NE. 0) GO TO 3560
      DO 3545 I=1,9
      SMC(L,I) = C(I)
 3545 IF(L .EQ. 2 .OR. L .EQ. 4) SMC(L,I)=-C(I)
 3550 CONTINUE
```

```
      RETURN
 3560 N1 = CYCLE - 1
      PRINT 3564,N1
 3564 FORMAT( /// ' **** ERROR IN SUBROUTINE LSQPF1 (STMT 3540).  SURFAC
     1E TENSION EFFECT IS NEGLECTED AFTER CYCLE' I4)
      LERROR = 1
      RETURN
      END
'FOR,IS S06,S06
      SUBROUTINE ARROW(IX1,IY1,IX2,IY2,IHITE,IBASE)
      HEIGHT = IHITE
      BASE = IBASE
      X2     = IX2
      Y2     = IY2
      DY     = IY2 - IY1
      DX     = IX2 - IX1
      SQ = 1./SQRT(DX*DX + DY*DY)
      FACTX = BASE * (SQ*DX)
      FACTY = BASE * (SQ*DY)
      X3 = X2 - HEIGHT * (SQ*DX)
      Y3 = Y2 - HEIGHT * (SQ*DY)
      IX4 = (X3 + FACTY) + .5
      IY4 = (Y3 - FACTX) + .5
      IX5 = (X3 - FACTY) + .5
      IY5 = (Y3 + FACTX) + .5
      CALL LINEV (IX4,IY4,IX2,IY2)
      CALL LINEV (IX5,IY5,IX2,IY2)
      RETURN
      END
'FOR,IS S07,S07
      SUBROUTINE FCTN1(K,X,L,M)
      COMMON/L3/ SP(482), DS, STC, STH, STR, STZ, STR2, STZ2, STRZ2,
     1  STZR2, NSP, LSEG(5), LSEGS(5), SMC(4,9)
        = 0
       4 - K
          430 I=1,4
          N4 + 2
           (3810,3815), M
 38.0 Z   (SP(N4+1) - SP(N4-1))/(SP(N4) - SP(N4-2))
      IF(Z .LT. X) RETURN
      GO TO 3830
 3815 Z = (SP(N4) - SP(N4-2))/(SP(N4+1)-SP(N4-1))
      IF(Z .GT. X) RETURN
 3830 CONTINUE
      L = 1
      RETURN
      ENTRY FCTN2(X,XTE,Y)
      X = SQRT(STR2-STRZ2*(Y-STC)**2)
      X1 = 0.5*XTE
      Y1 = 0.5001*XTE
 3840 IF(ABS(X-X1) .LT. Y1) GO TO 3850
      X1 = X1 + XTE
      GO TO 3840
 3850 X = X1
      RETURN
      END
'FOR,IS S09,S09
      SUBROUTINE FORCE(K,N1,N2,N3,N4,N5,STU,STV)
```

```
C **** FORCE IS COMPUTED ONLY WHEN SP(K) CROSSES XC OR YC ONCE.
      DIMENSION SMX(3), SMY(3)
      COMMON/L3/ SP(482), DS, STC, STH, STR, STZ, STR2, STZ2, STRZ2,
     1  STZR2, NSP, LSEG(5), LSEGS(5), SMC(4,9)
      N4 = 0
      N5 = 0
      X = FLOAT(N1) - 1.5
      Y = FLOAT(N2) - 1.5
      HMX = X - 0.5
      HPX = X + 0.5
      HMY = Y - 0.5
      HPY = Y + 0.5
      IF(K .GT. LSEGS(N3+1)) N3=N3+1
      N9 = LSEG(N3)
      GO TO (3710,3730,3710,3730), N9
 3710 SMX(1) = X - 0.5
      SMX(2) = X
      SMX(3) = X + 0.5
      DO 3715 M=1,3
      SMY(M) = SMC(N9,1)
      DO 3715 N=1,7
 3715 SMY(M) = SMY(M) + SMC(N9,N+1)*SMX(M)**N
      IF(SMY(2) .LT. HPY .AND. SMY(2) .GT. Y) N5=N2
      IF(SMY(2) .GT. HMY .AND. SMY(2) .LE. Y) N5=N2-1
      IF(SMY(1) .LT. Y .AND. SMY(2) .GT. Y .AND. SMY(3) .GT. Y) N4=N1-1
      IF(SMY(1) .GT. Y .AND. SMY(2) .LT. Y .AND. SMY(3) .LT. Y) N4=N1-1
      IF(SMY(1) .GT. Y .AND. SMY(2) .GT. Y .AND. SMY(3) .LT. Y) N4=N1
      IF(SMY(1) .LT. Y .AND. SMY(2) .LT. Y .AND. SMY(3) .GT. Y) N4=N1
      IF(N4 .EQ. 0 .AND. N5 .EQ. 0) RETURN
      XC = (SMY(3) - SMY(1)) / (SMX(3) - SMX(1))
      YC = SMY(1) - XC*SMX(1)
      YD = Y + X/XC
      GO TO 3740
 3730 SMY(1) = Y - 0.5
      SMY(2) = Y
      SMY(3) = Y + 0.5
      DO 3735 M=1,3
      SMX(M) = SMC(N9,1)
      DO 3735 N=1,7
 3735 SMX(M) = SMX(M) + SMC(N9,N+1)*SMY(M)**N
      IF(SMX(2) .LT. HPX .AND. SMX(2) .GT. X) N4=N1
      IF(SMX(2) .GT. HMX .AND. SMX(2) .LE. X) N4=N1-1
      IF(SMX(1) .LT. X .AND. SMX(2) .GT. X .AND. SMX(3) .GT. X) N5=N2-1
      IF(SMX(1) .GT. X .AND. SMX(2) .LT. X .AND. SMX(3) .LT. X) N5=N2-1
      IF(SMX(1) .GT. X .AND. SMX(2) .GT. X .AND. SMX(3) .LT. X) N5=N2
      IF(SMX(1) .LT. X .AND. SMX(2) .LT. X .AND. SMX(3) .GT. X) N5=N2
      IF(N4 .EQ. 0 .AND. N5 .EQ. 0) RETURN
      XC = (SMX(3) - SMX(1))/(SMY(3) - SMY(1))
      YC = SMX(1) - XC*SMY(1)
      YD = X + Y/XC
 3740 X = XC*(YD - YC)/(1.0 + XC**2)
      Y = SMC(N9,2)
      Z = 0.0
      DO 3750 N=1,6
      Y = Y + FLOAT(N+1)*SMC(N9,N+2)*X**N
 3750 Z = Z + FLOAT(N*(N+1))*SMC(N9,N+2)*X**(N-1)
      TEMP2 = (1.0 + Y**2)**2
      GO TO (3760,3770,3760,3770), N9
```

```
 3760 TEMP1 = X*(1.0 + Y**2)
      STV = Z/TEMP2 + Y/TEMP1
      STU = -Y*Z/TEMP2 - Y**2/TEMP1
      RETURN
 3770 TEMP1 = (YD - X/XC)*(1.0 + Y**2)
      STV = -Y*Z/TEMP2 + Y/TEMP1
      STU = Z/TEMP2 - 1.0/TEMP1
      RETURN
      END
@MAP,IL MAIN,MAIN
 LIB  SYS$*MSFC$.
 IN   S01
@COPOUT TPF$.,LHMAC2.
@REWIND,I LHMAC2.
@FREE LHMAC2.
@XQT  MAIN
C
C   **** DATA DECK ****
C
@FIN
@FIN
```

Appendix B

LISTING OF LHMAC 3 PROGRAM

Listing of LHMAC3 Program

```
@RUN,//T
@MSG,N LOCKHEED-HUNTSVILLE 3D MAC PROGRAM (TAPE GEN 090272)
@ASG,T LHMAC3,T,SAVE05 . LOCKHEEDMAC3PROGRAM
@REWIND LHMAC3.
@FOR,IS S01,S01
C
C LOCKHEED/HUNTSVILLE 3D MAC PROGRAM (LHMAC3, 64K CORE SPACE, 4 DRUMS)
C
      COMMON/L1/ ITYPE, IBAR, JBAR, KBAR, IP2, JP2, KP2, IARRAY(5000),
     1  ARRAY1(5000), ARRAY2(5000), ARRAY3(5000), ARRAY4(5000),
     2 ARRAY5(5000)
      EQUIVALENCE (IARRAY,IFLAG), (ARRAY1,PHI), (ARRAY2,U), (ARRAY3,V),
     1  (ARRAY4,W), (ARRAY5,D)
      INTEGER TYPE(22)
      DATA TYPE/12H 9 IN PAPER , 20*6H      /
   50 FORMAT(16I5)
      CALL IDENT(9,TYPE)
   55 READ 50, ITYPE, IBAR, JBAR, KBAR
      IP2 = IBAR + 2
      JP2 = JBAR + 2
      KP2 = KBAR + 2
      IF(ITYPE) 75,65,65
   65 CALL MAIN(IFLAG,PHI,U,V,W,D)
      GO TO 55
   75 CALL ENDJOB
      STOP
      END
@FOR,IS S02,S02
      SUBROUTINE MAIN(IFLAG,PHI,U,V,W,D)
      COMMON/L1/ ITYPE, IBAR, JBAR, KBAR, IP2, JP2, KP2, IARRAY(5000),
     1  ARRAY1(5000), ARRAY2(5000), ARRAY3(5000), ARRAY4(5000),
     2 ARRAY5(5000)
      COMMON/L2/ IX(100), IZ(100), X(2000), Y(2000), Z(2000), PLTX,
     1  PLTY, RPPUL, MARGNX, MARGNZ, NSEGPT, NIMRKR, COSPSI, SINPSI,
     2  CONTR1, CONTR2, CONTR3, CONTR4, XV(3,31), ZV(3,31)
      COMMON/L3/ IE, IEH, IS, ISH, IFUL, IFH, IB, ICST1, ICST3, IP1,
     1  JP1, KP1, ICYCLE, ICELL(6), KLMT
      COMMON/L4/ DXODY, DXODZ, DZODX, DZODY, DYODX, DYODZ
      DIMENSION IFLAG(KP2,JP2,IP2), PHI(KP2,JP2,IP2), U(KP2,JP2,IP2),
     1  V(KP2,JP2,IP2), W(KP2,JP2,IP2), D(KP2,JP2,IP2)
      DIMENSION TITLE(12), IOPT(16), IPLT(16), IPRT(16), GRT(16),
     1  GRX(30), GRY(30), GRZ(30), BDRY(6), NSEGV(3), JPLANE(3)
      DATA IE,IEH,IS,ISH,IFUL,IFH,IB/1000,2000,3000,4000,5000,6000,
     1  7000/, ICST1,ICST2,ICST3,ICST4,ICST5,ICST6/10,100,1000,2000,
     2  10000,5808/, CST1,CST2,CST3/0.0,0.0,0.0/,
     3  PI,PSI/3.1415927,1.0472/, ISIGN/-1/
  100 FORMAT(12A6)
  110 FORMAT(16I5)
  114 FORMAT(16F5.1)
  120 FORMAT(8F10.4)
  140 FORMAT(1H1, 12A6)
  150 FORMAT(// (16I8))
  154 FORMAT(/ (16F8.2))
  160 FORMAT(// (26I5))
  170 FORMAT(// (10E13.6))
  174 FORMAT(// ( 8F16.8))
  180 FORMAT(3I5, I10, 5E20.8)
  190 FORMAT( /)
```

```
      NU1 = 20
      NU2 = 21
      NU3 = 22
      NU4 = 23
      REWIND  11
      READ 100, TITLE
      READ 110, LNTH1, LNTH2, LNTH3, LNTH4, LNTH5, LNTH6, LNTH7
      READ 110, NMPPUX, NMPPUY, NMPPUZ
      READ 110, (IOPT(I),I=1,16), (IPLT(I),I=1,16), (IPRT(I),I=1,16)
      READ 110, NGRT, LHT, NVPLT, (NSEGV(I),I=1,3), (JPLANE(I),I=1,3)
      DO 251 J=1,NVPLT
      K = NSEGV(J)
      L = J
      READ 114, (XV(L,I),I=1,K)
      READ 114, (ZV(L,I),I=1,K)
      XV(L,K+1) = XV(L,1)
  251 ZV(L,K+1) = ZV(L,1)
      NMPPC = NMPPUX*NMPPUY*NMPPUZ
      J1 = NGRT + 1
      J2 = 2*NGRT
      READ 120, (BDRY(I),I=1,6)
      READ 120, (GRT(I),I=1,J1)
      READ 120, (GRX(I),I=1,J2)
      READ 120, (GRY(I),I=1,J2)
      READ 120, (GRZ(I),I=1,J2)
      READ 120, DT, DBETA, DX, DY, DZ, EPSA, EPSD, EPSP, EPSV, RHO, RNU,
     1  VSCALE, WALL
      READ 120, TIN, TPLT, TPRT, TCOMP, TFIN, TCPU
      IF(IPRT(1) .EQ. 0) GO TO 391
      PRINT 140, TITLE
      CALL CLKOUT(TCPU)
      PRINT 310, (IOPT(I),I=1,16), (IPLT(I),I=1,16), (IPRT(I),I=1,16)
  310 FORMAT(// ' VALUES OF IOPT(I), IPLT(I) AND IPRT(I)'// (16I8))
      PRINT 314, ITYPE, IBAR, JBAR, KBAR, LNTH1, LNTH2, LNTH3, LNTH4,
     1  LNTH5, LNTH6, LNTH7, NMPPUX, NMPPUY, NMPPUZ, NMPPC
  314 FORMAT(// ' ITYPE =' I2, 4X, 'IBAR =' I3, 4X, 'JBAR =' I3, 4X,
     1  'KBAR =' I3, 4X, 'LNTH1 =' I3, '    LNTH2 =' I3, '    LNTH3 =' I3,
     2  '    LNTH4 =' I3, '    LNTH5 =' I3, '    LNTH6 =' I3// ' LNTH7 ='
     3  I3, '    NMPPUX =' I2, '    NMPPUY =' I2, '    NMPPUZ =' I2,
     4  '    NMPPC =' I3)
      PRINT 318, NGRT, LHT, NVPLT, (NSEGV(I),I=1,3), (JPLANE(I),I=1,3)
  318 FORMAT(// ' NGRT =' I3, 4X, 'LHT =' I3, 5X, 'NVPLT =' I2,4X,
     1  'NSEGV(I) =' 3I3, 7X, 'JPLANE(I) =' 3I3)
      PRINT 322, (GRT(I),I=1,J1)
  322 FORMAT(// ' VALUES OF GRT(I), GRX(J), GRY(J), GRZ(J) AND BDRY(I)'
     1 // (10F13.6))
      PRINT 170, (GRX(I),I=1,J2)
      PRINT 170, (GRY(I),I=1,J2)
      PRINT 170, (GRZ(I),I=1,J2)
      PRINT 170, (BDRY(I),I=1,6)
      PRINT 326, DT, DBETA, DX, DY, DZ, EPSA, EPSD, EPSP, EPSV, RHO,
     1  RNU, VSCALE, WALL
  326 FORMAT(// ' DT =' F10.7, '    DBETA =' F7.3, '    DX =' F7.4, 5X,
     1  'DY =' F7.4, 5X, 'DZ =' F7.4, 5X, 'EPSA =' F6.4, 4X, 'EPSD ='
     2  F6.3, 4X, 'EPSP =' F6.4// ' EPSV =' F5.2, 5X, 'RHO =' F8.2, 3X,
     3  'RNU =' F10.7, '  VSCALE =' F6.3, '    WALL =' F4.1)
      PRINT 330, TIN, TPLT, TPRT, TCOMP, TFIN, TCPU
  330 FORMAT(// ' TIN =' F6.3, 5X, 'TPLT =' F6.3, 4X, 'TPRT =' F8.5, 2X,
```

```
     1   'TCOMP =' F8.5, ' TEIN =' F8.4, '   TCPU =' F8.1///
     2   ' VALUES OF XV(J,I) AND ZV(J,I)'/)
      DO 335 J=1,NVPLT
      K = NSEGV(J) + 1
      L = J
      PRINT 154, (XV(L,I),I=1,K)
  335 PRINT 154, (ZV(L,I),I=1,K)
  391 CONTINUE
      J1 = IP2*JP2*KP2
      IF(J1 .LE. ICST6) GO TO 405
      PRINT 400, IP2, JP2, KP2, J1
  400 FORMAT(// ' ** ERROR **  EXECUTION IS TERMINATED DUE TO SIZE LIMIT
     1ATION(400).  VALUES OF IP2, JP2, KP2 AND THEIR PRODUCT ARE' 3I4,
     2  I6, '.')
      RETURN
  405 CONTINUE
      COSPSI = COS(PSI)
      SINPSI = SIN(PSI)
      TEMP1 = LNTH1
      TEMP2 = LNTH2
      TEMP3 = LNTH3
      GO TO (511,521,521,521), ITYPE
  511 J1 = TEMP1 + TEMP2*COSPSI + 5.0
      J2 = TEMP3 + TEMP2*SINPSI + 1.0
      GO TO 571
  521 CONTINUE
      PRINT 564
  564 FORMAT(// ' ** ERROR **  THE MARGINS OF A PLOT HAS NOT BEEN DEFINE
     1D YET(571).')
      RETURN
C * COMP RSTR PTS FOR PLTS.  * ASG MRKR PARTICLES.
  571 NRPPUL = 1023/MAX0(J1,J2)
      RPPUL = NRPPUL
      MARGNX = (1023 - J1*NRPPUL)/2
      MARGNZ = (1023 - J2*NRPPUL)/2
      GO TO (581,701,701,701), ITYPE
  581 J1 = TEMP2*COSPSI*RPPUL
      J2 = TEMP2*SINPSI*RPPUL
      J3 = LNTH1*NRPPUL
      J4 = LNTH3*NRPPUL
      VPMGX = FLOAT(J1)/2.0
      VPMGZ = FLOAT(J2)/2.0
      DO 585 I=1,NVPLT
      K = NSEGV(I) + 1
      DO 585 J=1,K
      XV(I,J) = RPPUL*XV(I,J) + VPMGX
  585 ZV(I,J) = RPPUL*ZV(I,J) + VPMGZ
      N = 1
      DO 591 I=1,2
      IX(N) = MARGNX
      IX(N+1) = MARGNX + J3
      IX(N+2) = IX(N+1) + J1
      IX(N+3) = IX(N+2) - J3
      IX(N+4) = IX(N)
      IZ(I) = MARGNZ
      IZ(I+2) = MARGNZ + J2
      IZ(I+5) = MARGNZ + J4
      IZ(I+7) = MARGNZ + J4 + J2
```

```
591 N = N + 5
    K = 7
    M = 6
    DO 601 I=11,12
    IX(I) = IX(K+2)
    IX(I+2) = IX(K+1)
    IX(I+4) = IX(K)
    IZ(M) = IZ(M-4)
    IZ(I) = IZ(K+2)
    IZ(I+2) = IZ(K+1)
    IZ(I+4) = IZ(K)
    K = K - 6
601 M = M + 5
    NSEGPT = 16
701 CONTINUE
    IF(IPRT(2) .EQ. 0) GO TO 711
    PRINT 704, NSEGPT
704 FORMAT(// ' NSEGPT =' I3, '  AND VALUES OF IX(I) AND IZ(I) ARE GIV
   1EN BELOW.')
    PRINT 150, (IX(I),I=1,NSEGPT)
    PRINT 150, (IZ(I),I=1,NSEGPT)
711 CONTINUE
    IP1 = IBAR + 1
    JP1 = JBAR + 1
    KP1 = KBAR + 1
    CONTR1 = IBAR
    CONTR2 = JBAR
    CONTR3 = KBAR
    CONTR4 = 0.0
    DO 741 K=1,KP2
    DO 741 J=1,JP2
    DO 741 I=1,IP2
    IFLAG(K,J,I) = IF
    PHI(K,J,I) = 0.0
    U(K,J,I) = 0.0
    V(K,J,I) = 0.0
    W(K,J,I) = 0.0
741 D(K,J,I) = 0.0
    TEMP1 = 1.0/FLOAT(NMPPUX)
    TEMP2 = 1.0/FLOAT(NMPPUY)
    TEMP3 = 1.0/FLOAT(NMPPUZ)
    XC = IBAR
    YC = JBAR
    ZC = LHT
    XX = 0.5*TEMP1
    YY = 0.5*TEMP2
    ZZ = 0.5*TEMP3
    PLTY = TEMP2
    NMARKR = 0
    M = 0
    GO TO (751,1101,1101,1101), ITYPE
751 TEMP4 = ZC - TEMP3
    PLTX = XC - TEMP1
755 K = 2.0 + ZZ
761 J = 2.0 + YY
765 I = 2.0 + XX
    NMARKR = NMARKR + 1
    M = M + 1
```

```
      X(M) = XX
      Y(M) = YY
      Z(M) = ZZ
      IF(M .NE. ICST4) GO TO 771
      WRITE(NU1) (X(L),L=1,ICST4),(Y(L),L=1,ICST4), (Z(L),L=1,ICST4)
      M = 0
  771 CONTINUE
      IFLAG(K,J,I) = IFUL + 1
      XX = XX + TEMP1
      IF(XX .LT. XC) GO TO 765
      XX = 0.5*TEMP1
      YY = YY + TEMP2
      IF(YY .LT. YC) GO TO 761
      XX = 0.5*TEMP1
      YY = 0.5*TEMP2
      ZZ = ZZ + TEMP3
      IF(ZZ .LT. TEMP4 .OR. ZZ .GT. ZC) GO TO 775
      NIMPKP = NMARKP
      XX = 0.5*XX
      YY = 0.5*YY
      TEMP1 = 0.5*TEMP1
      TEMP2 = 0.5*TEMP2
  775 IF(ZZ .LT. ZC) GO TO 755
      IF(M .LT. 1 ) GO TO 781
      WRITE(NU1) (X(L),L=1,M), (Y(L),L=1,M), (Z(L),L=1,M)
  781 CONTINUE
      DO 785 J=2,JP1
      DO 785 I=2,IP1
      IFLAG(1,J,I) = IB
  785 IFLAG(KP2,J,I) = IB
      DO 791 K=2,KP1
      DO 791 I=2,IP1
      IFLAG(K,1,I) = IB
  791 IFLAG(K,JP2,I) = IB
      DO 795 K=2,KP1
      DO 795 J=2,JP1
      IFLAG(K,J,1) = IB
  795 IFLAG(K,J,IP2) = IB
 1191 CONTINUE
      DXODY = DX/DY
      DXODZ = DX/DZ
      DZODX = DZ/DX
      DZODY = DZ/DY
      DYODX = DY/DX
      DYODZ = DY/DZ
      DXDX = DX*DX
      DYDY = DY*DY
      DZDZ = DZ*DZ
      DXDY = DX*DY
      DYDZ = DY*DZ
      DZDX = DZ*DX
      COE1 = 2.0*RNU/DX
      COE2 = 2.0*RNU/DY
      COE3 = 2.0*RNU/DZ
      COE4 = (1.0 + EPSA)/(2.0/DXDX + 2.0/DYDY + 2.0/DZDZ)
      ITEMP = IPRT(5)
      KLMT = LHT + 2
      KLIMIT = KLMT
```

```
      REWIND NU4
      WRITE (NU4) (((D(K,J,I),I=1,IP2),J=1,JP2),K=1,KLIMIT)
      CALL CELLID(IFLAG,PHI,U,V,W,D,ITEMP)
      ICYCLE = 0
      T = 0.0
      JCMP = 0
      JPLT = 0
      JPRT = 0
      INDU13 = 1
 1001 KPLT = T/TPLT
      KPRT = T/TPRT
      KCMP = T/TCOMP
      IF(KPLT .NE. JPLT) GO TO 1591
      IF(IPLT(1) .EQ. 0) GO TO 1491
      I2 = ITYPE
      I3 = IOPT(1)
      I4 = 1
      REWIND NU1
      M = NMARKP
      CALL FRAMEV(3)
 1421 IF(M .GT. ICST4) M=ICST4
      READ(NU1) (X(L),L=1,M), (Y(L),L=1,M), (Z(L),L=1,M)
      CALL PLOT (1,ITYPE,I3,I4,M)
      M = NMARKP - I4*ICST4
      IF(M .LT. 1) GO TO 1491
      I4 = I4 + 1
      GO TO 1421
 1491 CONTINUE
      IF(IPLT(2) .EQ. 0 .OR. KPLT .EQ. 0) GO TO 1581
      NSTMT = 1521
      DO 1521 L=1,NVPLT
      CALL FRAMEV(3)
      J = JPLANE(L)
      I4 = NSEGV(L)
      M = 0
      DO 1515 K=2,KLMT
      ZZ = RPPUL*(FLOAT(K) - 1.5) + VPMGZ
      DO 1511 I=2,IP1
      IF(IFLAG(K,J,I) .LT. IS) GO TO 1511
      M = M + 1
      X(M) = RPPUL*(FLOAT(I) - 1.5) + VPMGX
      Z(M) = ZZ
      X(ICST3+M) = 0.5*RPPUL*VSCALE*(U(K,J,I-1) + U(K,J,I)) + X(M)
      Z(ICST3+M) = 0.5*RPPUL*VSCALE*(W(K-1,J,I) + W(K,J,I)) + Z(M)
 1511 CONTINUE
 1515 CONTINUE
      IF(M .GT. ICST3) GO TO 3911
      CALL PLOT (2,ICST3,L,I4,M)
 1521 CONTINUE
 1581 JPLT = JPLT + 1
 1591 CONTINUE
      IF(IPRT(3) .EQ. 0) GO TO 1891
      IF(KPRT .NE. JPRT) GO TO 1891
      REWIND NU4
      READ (NU4) (((D(K,J,I),I=1,IP2),J=1,JP2),K=1,KLIMIT)
      DO 1785 K=2,KLMT
      DO 1785 J=2,JP1
      DO 1781 I=2,IP1
```

```
      IF(IFLAG(K,J,I) .LT. IFULL) GO TO 1781
      D(K,J,I) = D(K,J,I)
     1  - (PHI(K,J,I+1) + PHI(K,J,I-1) - 2.0*PHI(K,J,I))/DXDX
     2  - (PHI(K,J+1,I) + PHI(K,J-1,I) - 2.0*PHI(K,J,I))/DYDY
     3  - (PHI(K+1,J,I) + PHI(K-1,J,I) - 2.0*PHI(K,J,I))/DZDZ
 1781 CONTINUE
 1785 CONTINUE
      PRINT 1810, T, GX, GY, GZ, ICYCLE, ITER
 1810 FORMAT(1H1, ' VELOCITIES AND PRESSURE AT TIME' F8.4, ' SEC, GX =' 
     1  F8.4, ' GY =' F8.4, ' GZ =' F8.4, ' ICYCLE =' I4, ' ITER =' I5//
     2  ' K  J  I  IFLAG    U' 9X, 'V' 9X, 'W' 9X, 'D' 9X, 'PHI' 8X,
     3  ' I  IFLAG    U' 9X, 'V' 9X, 'W' 9X, 'D' 9X, 'PHI')
      DO 1875 K=1,KLMT
      DO 1875 J=1,JP2
      DO 1871 I=1,IP2,2
      L = I + 1
      IF(L .GT. IP2) GO TO 1861
      IF(I .EQ. 1) PRINT 190
      PRINT 1830, K, J, I, IFLAG(K,J,I), U(K,J,I), V(K,J,I), W(K,J,I),
     1  D(K,J,I), PHI(K,J,I), L, IFLAG(K,J,L), U(K,J,L), V(K,J,L),
     2  W(K,J,L), D(K,J,L), PHI(K,J,L)
 1830 FORMAT(3I3, I6, 5F10.4, 2I6, 5F10.4)
      GO TO 1871
 1861 PRINT 1830, K, J, I, IFLAG(K,J,I), U(K,J,I), V(K,J,I), W(K,J,I),
     1  D(K,J,I), PHI(K,J,I)
 1871 CONTINUE
 1875 CONTINUE
      JPRT = JPRT + 1
 1891 CONTINUE
      IF(KCMP .NE. JCMP) GO TO 1991
      FX = 0.0
      FY = 0.0
      FZ = 0.0
      RMX = 0.0
      RMY = 0.0
      RMZ = 0.0
      I2 = 2
      GO TO (1911,1911,1981,1981), ITYPE
 1911 DO 1931 K=2,KLMT
      TEMP3 = K
      TEMP3 = DZ*(TEMP3 - 1.5)
      DO 1921 J=2,JP1
      TEMP2 = - JBAR/2 + J
      TEMP2 = DY*(TEMP2 - 1.5)
      TEMP4 = RHO*DYDZ*(PHI(K,J,IP1) - PHI(K,J,I2))
      FX = FX + TEMP4
      RMY = RMY + TEMP3*TEMP4
 1921 RMZ = RMZ - TEMP2*TEMP4
      DO 1925 I=2,IP1
      TEMP1 = - IBAR/2 + I
      TEMP1 = DX*(TEMP1 - 1.5)
      TEMP5 = RHO*DZDX*(PHI(K,JP1,I) - PHI(K,I2,I))
      FY = FY + TEMP5
      RMX = RMX - TEMP3*TEMP5
 1925 RMZ = RMZ + TEMP1*TEMP5
 1931 CONTINUE
      DO 1941 I=2,IP1
      TEMP1 = - IBAR/2 + I
```

```
      TEMP1 = DX*(TEMP1 - 1.5)
      DO 1041 J=2,JP1
      TEMP2 = - JBAR/2 + J
      TEMP2 = DY*(TEMP2 - 1.5)
      TEMP6 = RHO*DXDY*(PHI(KP1,J,I) - PHI(12,J,I))
      FZ = FZ + TEMP6
      RMX = RMX + TEMP2*TEMP6
 1041 RMY = RMY - TEMP1*TEMP6
      DO 1051 K=2,KLMT
      DO 1051 J=2,JP1
      DO 1051 I=2,IP1
 1051 D(K,J,I) = RHO*PHI(K,J,I)
      L = 2
      PRINT 1970
 1970 FORMAT(//// ' PRESSURE DISTRIBUTION(N/M**2) ON THE LEFT(K,J,2), RI
     1GHT(K,J,IP1), FRONT(K,2,I), BACK(K,JP1,I), BOTTOM(2,J,I) AND TOP(K
     2P1,J,I) '/ '   WALLS OF THE CONTAINER, RESPECTIVELY ' )
      PRINT 190
      DO 1071 J=2,JP1
      N = J
 1071 PRINT 1972, (D(K,J,L),K=2,KLMT)
 1972 FORMAT(10F13.6)
      PRINT 190
      DO 1073 J=2,JP1
      N = J
 1073 PRINT 1972, (D(K,J,IP1),K=2,KLMT)
      PRINT 190
      DO 1074 K=2,KLMT
      N = K
 1074 PRINT 1972, (D(N,L,I),I=2,IP1)
      PRINT 190
      DO 1075 K=2,KLMT
      N = K
 1075 PRINT 1972, (D(N,JP1,I),I=2,IP1)
      PRINT 190
      DO 1076 J=2,JP1
      N = J
 1076 PRINT 1972, (D(L,N,I),I=2,IP1)
      PRINT 190
      DO 1077 J=2,JP1
      N = J
 1077 PRINT 1972, (D(KP1,N,I),I=2,IP1)
 1081 JCMP = JCMP + 1
      PRINT 1984, T, FX, FY, FZ, RMX, RMY, RMZ
 1984 FORMAT(// ' T =' F10.5, ' (SEC), FX =' F10.4, ', FY =' F10.4,
     1  ', FZ =' F10.4, ' (N), RMX =' F10.4, ', RMY =' E10.4, ', RMZ ='
     2  F10.4, ' (N-M).')
 1991 CONTINUE
C *CK IF BDRY PRESSURE NEEDS TO BE ADJUSTED.
      DO 2155 K=2,KLMT
      DO 2155 J=2,JP1
      DO 2151 I=2,IP1
      IF(IFLAG(K,J,I) .GT. ISH) GO TO 2151
      IF(IFLAG(K,J,I) .LT. IS) GO TO 2141
      M = (IFLAG(K,J,I) - IS)/ICST1
      IF(M .GT. ICST3) M=M-ICST3
      IF(M .EQ. 1  .OR. M .EQ. 4 ) GO TO 2111
      IF(M .EQ. 2  .OR. M .EQ. 8 ) GO TO 2121
```

```
      IF(M .EQ. 16 .OR. M .EQ. 32) GO TO 2131
      GO TO 2141
 2111 PHI(K,J,I) = COE1*(U(K,J,I) - U(K,J,I-1))
      GO TO 2151
 2121 PHI(K,J,I) = COE3*(W(K,J,I) - W(K-1,J,I))
      GO TO 2151
 2131 PHI(K,J,I) = COE2*(V(K,J,I) - V(K,J-1,I))
      GO TO 2151
 2141 PHI(K,J,I) = 0.0
 2151 CONTINUE
 2155 CONTINUE
      ASSIGN 2185 TO ME
 2161 L1 = 1
      DO 2171 J=2,JP1
      DO 2171 I=2,IP1
      PHI(L1,J,I) = PHI(L1+1,J,I)
 2171 PHI(KP2,J,I) = PHI(KP1,J,I)
      DO 2175 K=2,KLMT
      DO 2175 I=2,IP1
      PHI(K,L1,I) = PHI(K,L1+1,I)
 2175 PHI(K,JP2,I) = PHI(K,JP1,I)
      DO 2181 K=2,KLMT
      DO 2181 J=2,JP1
      PHI(K,J,L1) = PHI(K,J,L1+1)
 2181 PHI(K,J,IP2) = PHI(K,J,IP1)
      GO TO ME
 2185 CONTINUE
      ICYCLE = ICYCLE + 1
      T = T + DT
      IF(T .GT. TFIN) RETURN
      REWIND NU2
      WRITE(NU2) (((PHI(K,J,I),I=1,IP2),J=1,JP2),K=1,KLMT)
      DO 2191 I=1,NGRT
      IF(T .GT. GRT(I+1)) GO TO 2191
      J = 2*I - 1
      TEMP1 = (T - GRT(I))/(GRT(I+1) - GRT(I))
      GX = GRX(J) + TEMP1*(GRX(J+1) - GRX(J))
      GY = GRY(J) + TEMP1*(GRY(J+1) - GRY(J))
      GZ = GRZ(J) + TEMP1*(GRZ(J+1) - GRZ(J))
      GO TO 2195
 2191 CONTINUE
 2195 CONTINUE
      DO 2233 K=2,KLMT
      DO 2233 J=2,JP1
      DO 2231 I=2,IP1
      IF(IFLAG(K,J,I) .LT. 15) GO TO 2231
      TEMP1 = U(K,J,I)*(U(K,J,I+1)-U(K,J,I-1))/DX
      TEMP2 = V(K,J,I)*(V(K,J+1,I)-V(K,J-1,I))/DY
      TEMP3 = W(K,J,I)*(W(K+1,J,I)-W(K-1,J,I))/DZ
      TEMP4 = (U(K,J+1,I)+U(K,J,I))*(V(K,J,I+1)+V(K,J,I))/4.0
      TEMP5 = (V(K+1,J,I)+V(K,J,I))*(W(K,J+1,I)+W(K,J,I))/4.0
      TEMP6 = (W(K,J,I+1)+W(K,J,I))*(U(K+1,J,I)+U(K,J,I))/4.0
      TEMP7 = (U(K,J,I)+U(K,J-1,I))*(V(K,J-1,I+1)+V(K,J-1,I))/4.0
      TEMP8 = (V(K,J,I)+V(K,J,I-1))*(U(K,J+1,I-1)+U(K,J,I-1))/4.0
      TEMP9 = (V(K,J,I)+V(K-1,J,I))*(W(K-1,J+1,I)+W(K-1,J,I))/4.0
      TEMP10= (W(K,J,I)+W(K,J-1,I))*(V(K+1,J-1,I)+V(K,J-1,I))/4.0
      TEMP11= (W(K,J,I)+W(K,J,I-1))*(U(K+1,J,I-1)+U(K,J,I-1))/4.0
      TEMP12= (U(K,J,I)+U(K-1,J,I))*(W(K-1,J,I+1)+W(K-1,J,I))/4.0
```

```
      TEMP13 = RNU*((U(K,J,I+1)+U(K,J,I-1)-2.0*U(K,J,I))/DXDX
     1  + (U(K,J+1,I)+U(K,J-1,I)-2.0*U(K,J,I))/DYDY
     2  + (U(K+1,J,I)+U(K-1,J,I)-2.0*U(K,J,I))/DZDZ) + GX
      TEMP14 = RNU*((V(K,J,I+1)+V(K,J,I-1)-2.0*V(K,J,I))/DXDX
     1  + (V(K,J+1,I)+V(K,J-1,I)-2.0*V(K,J,I))/DYDY
     2  + (V(K+1,J,I)+V(K-1,J,I)-2.0*V(K,J,I))/DZDZ) + GY
      TEMP15 = RNU*((W(K,J,I+1)+W(K,J,I-1)-2.0*W(K,J,I))/DXDX
     1  + (W(K,J+1,I)+W(K,J-1,I)-2.0*W(K,J,I))/DYDY
     2  + (W(K+1,J,I)+W(K-1,J,I)-2.0*W(K,J,I))/DZDZ) + GZ
      IF(K .EQ. KP1 .AND. IFLAG(K+1,J,I) .EQ. IB) GO TO 2211
      W(K,J,I) = W(K,J,I) + DT*(TEMP15 + (PHI(K,J,I) - PHI(K+1,J,I))/DZ
     1  - TEMP3 + (TEMP10 - TEMP5)/DY + (TEMP11 - TEMP6)/DX)
 2211 IF(J .EQ. JP1 .AND. IFLAG(K,J+1,I) .EQ. IB) GO TO 2221
      V(K,J,I) = V(K,J,I) + DT*(TEMP14 + (PHI(K,J,I) - PHI(K,J+1,I))/DY
     1  - TEMP2 + (TEMP9  - TEMP5)/DZ + (TEMP8  - TEMP4)/DX)
 2221 IF(I .EQ. IP1 .AND. IFLAG(K,J,I+1) .EQ. IB) GO TO 2231
      U(K,J,I) = U(K,J,I) + DT*(TEMP13 + (PHI(K,J,I) - PHI(K,J,I+1))/DX
     1  - TEMP1 + (TEMP12 - TEMP6)/DZ + (TEMP7  - TEMP4)/DY)
 2231 CONTINUE
 2233 CONTINUE
      ASSIGN 2281 TO ME
 2235 CALL VELCTY(IFLAG,PHI,U,V,W,D)
      L1 = 1
      N = 1
      DO 2241 J=2,JP1
      DO 2241 I=2,IP1
      IF(IFLAG(L1,J,I) .EQ. IB) U(L1,J,I)=BDRY(N)*U(L1+1,J,I)
      IF(IFLAG(L1,J,I) .EQ. IB) V(L1,J,I)=BDRY(N)*V(L1+1,J,I)
      IF(IFLAG(KP2,J,I) .EQ. IB) V(KP2,J,I)=BDRY(N+1)*V(KP1,J,I)
 2241 IF(IFLAG(KP2,J,I) .EQ. IB) U(KP2,J,I)=BDRY(N+1)*U(KP1,J,I)
      N = N + 2
      DO 2251 K=2,KLMT
      DO 2251 J=2,JP1
      IF(IFLAG(K,J,L1) .EQ. IB) V(K,J,L1)=BDRY(N)*V(K,J,L1+1)
      IF(IFLAG(K,J,L1) .EQ. IB) W(K,J,L1)=BDRY(N)*W(K,J,L1+1)
      IF(IFLAG(K,J,IP2) .EQ. IB) V(K,J,IP2)=BDRY(N+1)*V(K,J,IP1)
 2251 IF(IFLAG(K,J,IP2) .EQ. IB) W(K,J,IP2)=BDRY(N+1)*W(K,J,IP1)
      N = N + 2
      DO 2261 K=2,KLMT
      DO 2261 I=2,IP1
      IF(IFLAG(K,L1,I) .EQ. IB) U(K,L1,I)=BDRY(N)*U(K,L1+1,I)
      IF(IFLAG(K,L1,I) .EQ. IB) W(K,L1,I)=BDRY(N)*W(K,L1+1,I)
      IF(IFLAG(K,JP2,I) .EQ. IB) U(K,JP2,I)=BDRY(N+1)*U(K,JP1,I)
 2261 IF(IFLAG(K,JP2,I) .EQ. IB) W(K,JP2,I)=BDRY(N+1)*W(K,JP1,I)
      GO TO ME
 2281 DO 2295 K=2,KLMT
      DO 2295 J=2,JP1
      DO 2291 I=2,IP1
      IF(IFLAG(K,J,I) .LT. IFUL .OR. IFLAG(K,J,I) .GT. IFH) GO TO 2291
      D(K,J,I) = (U(K,J,I) - U(K,J,I-1))/DX + (V(K,J,I) - V(K,J-1,I))/DY
     1  + (W(K,J,I) - W(K-1,J,I))/DZ
      TEMP1 = U(K,J,I-2)*U(K,J,I-1)+U(K,J,I)*(U(K,J,I+1)-2.0*U(K,J,I-1))
      TEMP2 = V(K,J-2,I)*V(K,J-1,I)+V(K,J,I)*(V(K,J+1,I)-2.0*V(K,J-1,I))
      TEMP3 = W(K-2,J,I)*W(K-1,J,I)+W(K,J,I)*(W(K+1,J,I)-2.0*W(K-1,J,I))
      TEMP4 = (U(K,J,I) + U(K,J+1,I))*(V(K,J,I) + V(K,J,I+1))
     1  + (U(K,J-1,I-1) + U(K,J,I-1))*(V(K,J-1,I-1)+V(K,J-1,I))
     2  - (U(K,J-1,I) + U(K,J,I))*(V(K,J-1,I) + V(K,J-1,I+1))
     3  - (U(K,J,I-1) + U(K,J+1,I-1))*(V(K,J,I-1) + V(K,J,I))
```

```
      TEMP5 = (U(K,J,I) + U(K+1,J,I))*(W(K,J,I) + W(K,J,I+1))
     1  + (U(K-1,J,I-1) + U(K,J,I-1))*(W(K-1,J,I-1)+W(K-1,J,I))
     2  - (U(K-1,J,I) + U(K,J,I))*(W(K-1,J,I) + W(K-1,J,I+1))
     3  - (U(K,J,I-1) + U(K+1,J,I-1))*(W(K,J,I-1) + W(K,J,I))
      TEMP6 = (V(K,J,I) + V(K+1,J,I))*(W(K,J,I) + W(K,J+1,I))
     1  + (V(K-1,J-1,I) + V(K,J-1,I))*(W(K-1,J-1,I)+W(K-1,J,I))
     2  - (V(K-1,J,I) + V(K,J,I))*(W(K-1,J,I) + W(K-1,J+1,I))
     3  - (V(K,J-1,I) + V(K+1,J-1,I))*(W(K,J-1,I) + W(K,J,I))
      PHI(K,J,I) = D(K,J,I)/DT - TEMP1/DXDX - TEMP2/DYDY - TEMP3/DZDZ
     1  - 0.5*(TEMP4/DXDY + TEMP5/DZDX + TEMP6/DYDZ)
 2201 CONTINUE
 2205 CONTINUE
      DO 2305 K=2,KLMT
      DO 2305 J=2,JP1
      DO 2301 I=2,IP1
      IF(IFLAG(K,J,I) .LT. IFUL .OR. IFLAG(K,J,I) .GT. IFH) GO TO 2301
      PHI(K,J,I) = PHI(K,J,I) + RNU*((D(K,J,I+1) + D(K,J,I-1) - 2.0*D(
     1  K,J,I))/DXDX + (D(K,J+1,I) + D(K,J-1,I) - 2.0*D(K,J,I))/DYDY +
     2  (D(K+1,J,I) + D(K-1,J,I) - 2.0*D(K,J,I))/DZDZ)
 2301 CONTINUE
 2305 CONTINUE
      DO 2315 K=2,KLMT
      DO 2315 J=2,JP1
      DO 2311 I=2,IP1
      D(K,J,I) = 0.0
      IF(IFLAG(K,J,I) .LT. IFUL .OR. IFLAG(K,J,I) .GT. IFH) GO TO 2311
      D(K,J,I) = PHI(K,J,I)
 2311 CONTINUE
 2315 CONTINUE
      KLIMIT = KLMT
      REWIND NU4
      WRITE (NU4) (((D(K,J,I),I=1,IP2),J=1,JP2),K=1,KLIMIT)
      REWIND NU2
      READ(NU2) (((PHI(K,J,I),I=1,IP2),J=1,JP2),K=1,KLMT)
      ITER = 0
      ASSIGN 2411 TO ME
 2411 ITER = ITER + 1
      L = 0
      DO 2425 K=2,KLMT
      DO 2425 J=2,JP1
      DO 2421 I=2,IP1
      IF(IFLAG(K,J,I) .LT. IFUL .OR. IFLAG(K,J,I) .GT. IFH) GO TO 2421
      TEMP1 = ABS(PHI(K,J,I))
      PHI(K,J,I) = ((PHI(K,J,I+1) + PHI(K,J,I-1))/DXDX + (PHI(K,J+1,I)
     1  + PHI(K,J-1,I))/DYDY + (PHI(K+1,J,I) + PHI(K-1,J,I))/DZDZ
     2  - D(K,J,I))*COE4 - EPSA*PHI(K,J,I)
      IF(L .EQ. 1) GO TO 2421
      TEMP2 = ABS(PHI(K,J,I))
      TEMP3 = ABS(TEMP2 - TEMP1)/(TEMP2 + TEMP1)
      IF(TEMP3 .GT. EPSP) L=1
 2421 CONTINUE
 2425 CONTINUE
      IF(L .EQ. 0) GO TO 2431
      GO TO 2161
 2431 CALL CLKOUT(TCPU)
C *MV MARKER PARTICLES.
      REWIND NU1
      REWIND NU3
```

```
      M = NMARKP
      I4 = 1
 2521 IF(M .GT. ICST4) M=ICST4
      READ(NU1) (X(L),L=1,M), (Y(L),L=1,M), (Z(L),L=1,M)
      DO 2581 N=1,M
      I = X(N) + 2.0
      J = Y(N) + 2.0
      K = Z(N) + 2.0
      TEMP1 = I
      TEMP2 = J
      TEMP3 = K
      XC = TEMP1 - 1.5
      YC = TEMP2 - 1.5
      ZC = TEMP3 - 1.5
      DO 2571 L=1,3
      X1 = TEMP1 - 0.5 - X(N)
      Y1 = TEMP2 - 0.5 - Y(N)
      Z1 = TEMP3 - 0.5 - Z(N)
      I1 = I + 1
      J1 = J + 1
      K1 = K + 1
      GO TO (2531,2541,2551), L
 2531 I1 = I - 1
      X1 = X(N) + 2.0 - TEMP1
      X2 = 1.0 - X1
      IF(Y(N) .GT. YC) GO TO 2535
      J1 = J - 1
      Y1 = Y(N) + 2.5 - TEMP2
 2535 Y2 = 1.0 - Y1
      IF(Z(N) .GT. ZC) GO TO 2539
      K1 = K - 1
      Z1 = Z(N) + 2.5 - TEMP3
 2539 Z2 = 1.0 - Z1
      XX = DT*(Y1*Z1*(X1*U(K,J,I) + X2*U(K,J,I1)) + Y1*Z2*(X1*U(K1,J,I)
     1  + X2*U(K1,J,I1)) + Y2*Z1*(X1*U(K,J1,I) + X2*U(K,J1,I1))
     2  + Y2*Z2*(X1*U(K1,J1,I) + X2*U(K1,J1,I1)))
      GO TO 2571
 2541 J1 = J - 1
      Y1 = Y(N) + 2.0 - TEMP2
      Y2 = 1.0 - Y1
      IF(X(N) .GT. XC) GO TO 2545
      I1 = I - 1
      X1 = X(N) + 2.5 - TEMP1
 2545 X2 = 1.0 - X1
      IF(Z(N) .GT. ZC) GO TO 2549
      K1 = K - 1
      Z1 = Z(N) + 2.5 - TEMP3
 2549 Z2 = 1.0 - Z1
      YY = DT*(Z1*X1*(Y1*V(K,J,I) + Y2*V(K,J1,I)) + Z1*X2*(Y1*V(K,J,I1)
     1  + Y2*V(K,J1,I1)) + Z2*X1*(Y1*V(K1,J,I) + Y2*V(K1,J1,I))
     2  + Z2*X2*(Y1*V(K1,J,I1) + Y2*V(K1,J1,I1)))
      GO TO 2571
 2551 K1 = K - 1
      Z1 = Z(N) + 2.0 - TEMP3
      Z2 = 1.0 - Z1
      IF(X(N) .GT. XC) GO TO 2555
      I1 = I - 1
      X1 = X(N) + 2.5 - TEMP1
```

```
2555 X2 = 1.0 - X1
     IF(Y(N) .GT. YC) GO TO 2559
     J1 = J - 1
     Y1 = Y(N) + 2.5 - TEMP2
2559 Y2 = 1.0 - Y1
     ZZ = DT*(X1*Y1*(Z1*W(K,J,I) + Z2*W(K1,J,I)) + X1*Y2*(Z1*W(K,J1,I)
    1  + Z2*W(K1,J1,I)) + X2*Y1*(Z1*W(K,J,I1) + Z2*W(K1,J,I1))
    2  + X2*Y2*(Z1*W(K,J1,I1) + Z2*W(K1,J1,I1)))
2571 CONTINUE
     X(N) = X(N) + XX
     Y(N) = Y(N) + YY
2581 Z(N) = Z(N) + ZZ
     WRITE(NU3) (X(L),L=1,M), (Y(L),L=1,M), (Z(L),L=1,M)
     M = NMARKP - I4*ICST4
     IF(M .LT. 1) GO TO 2591
     I4 = I4 + 1
     GO TO 2521
2591 INDU13 = INDU13*ISIGN
     NU1 = NU2 - INDU13
     NU3 = NU2 + INDU13
     DO 2711 K=2,KP1
     DO 2711 J=2,JP1
     DO 2711 I=2,IP1
2711 D(K,J,I) = 0.0
     REWIND NU1
     M = NMARKP
     I4 = 1
     TEMP1 = IBAR
     TEMP2 = JBAR
     TEMP3 = KBAR
2721 IF(M .GT. ICST4) M=ICST4
     READ(NU1) (X(L),L=1,M), (Y(L),L=1,M), (Z(L),L=1,M)
     DO 2731 N=1,M
     I = X(N) + 2.0
     J = Y(N) + 2.0
     K = Z(N) + 2.0
     IF(I .LT. 1 .OR. I .GT. IP2) GO TO 2725
     IF(J .LT. 1 .OR. J .GT. JP2) GO TO 2725
     IF(K .LT. 1 .OR. K .GT. KP2) GO TO 2725
     IF(I .EQ. 1) X(N)=0.1
     IF(J .EQ. 1) Y(N)=0.1
     IF(K .EQ. 1) Z(N)=0.1
     IF(I .EQ. IP2) X(N)=TEMP1-0.1
     IF(J .EQ. JP2) Y(N)=TEMP2-0.1
     IF(K .EQ. KP2) Z(N)=TEMP3-0.1
     GO TO 2731
2725 PRINT 2726, ICYCLE, ITER, I, J, K, N, X(N), Y(N), Z(N)
2726 FORMAT(/// ' **** ERROR **** ' 6I8, 3F14.6)
     RETURN
2731 D(K,J,I) = 1.0
     M = NMARKP - I4*ICST4
     IF(M .LT. 1) GO TO 2741
     I4 = I4 + 1
     GO TO 2721
2741 DO 2755 K=2,KLMT
     DO 2755 J=2,JP1
     DO 2751 I=2,IP1
     N = D(K,J,I)
```

```
      IF(N .EQ. 0) GO TO 2751
      IFLAG(K,J,I) = IFLAG(K,J,I) + N
      J1 = K
 2751 CONTINUE
 2755 CONTINUE
      KLMT = J1 + 2
      IF(IPRT(5) .EQ. 1) PRINT 2770, ICYCLE, ITER
 2770 FORMAT(/// ' ICYCLE =' I3, 8X, 'ITER =' I5)
      IF(KLMT .GT. KP1) KLMT=KP1
      ITEMP = IPRT(5)
      CALL CELLID(IFLAG,PHI,U,V,W,D,ITEMP)
      ASSIGN 1201 TO ME
      GO TO 2235
 3911 PRINT 3914, NSTMT
 3914 FORMAT(// ' ** ERROR **  EXECUTION OF CURRENT CASE IS TERMINATED D
     1UE TO STORAGE LIMITATION(' I4, ').')
      RETURN
      END
'FOR,IS S03,S03
      SUBROUTINE CELLID(IFLAG,PHI,U,V,W,D,ITEMP)
      COMMON/L1/ ITYPE, IBAR, JBAR, KBAR, IP2, JP2, KP2, IARRAY(5000),
     1  ARRAY1(5000), ARRAY2(5000), ARRAY3(5000), ARRAY4(5000),
     2 ARRAY5(5000)
      COMMON/L3/ IE, IEH, IS, ISH, IFUL, IFH, IB, ICST1, ICST3, IP1,
     1  JP1, KP1, ICYCLE, ICELL(6), KLMT
      DIMENSION IFLAG(KP2,JP2,IP2), PHI(KP2,JP2,IP2), U(KP2,JP2,IP2),
     1  V(KP2,JP2,IP2), W(KP2,JP2,IP2), D(KP2,JP2,IP2)
      DATA I1,I2,I3,I4,I5,I6/1,2,3,4,5,6/
C *CK IF AN EMP CLL SHOULD BECOME A SUR CLL AND VICE VERSA.  *ADJ THE
C    VEL COMPS OF A NEWLY CREATED EMP CLL.
      DO 4111 K=2,KLMT
      DO 4111 J=2,JP1
      DO 4101 I=2,IP1
      M = IFLAG(K,J,I)/ICST3
      IF(M .GT. 4) GO TO 4101
      L = IFLAG(K,J,I)/ICST1
      L = IFLAG(K,J,I) - ICST1*L + 1
      GO TO (4021,4051), L
 4021 GO TO (4101,4101,4031,4041), M
 4031 IFLAG(K,J,I) = IE
      GO TO 4081
 4041 IFLAG(K,J,I) = IEH
      GO TO 4081
 4051 GO TO (4061,4071,4061,4071), M
 4061 IFLAG(K,J,I) = IS
      GO TO 4101
 4071 IFLAG(K,J,I) = ISH
      GO TO 4101
 4081 ICELL(I1)= IFLAG(K,J,I+1)/ICST3
      ICELL(I2)= IFLAG(K+1,J,I)/ICST3
      ICELL(I3)= IFLAG(K,J,I-1)/ICST3
      ICELL(I4)= IFLAG(K-1,J,I)/ICST3
      ICELL(I5)= IFLAG(K,J+1,I)/ICST3
      ICELL(I6)= IFLAG(K,J-1,I)/ICST3
      J1 = 0
      DO 4091 N=1,6
      IF(ICELL(N) .GT. 2) GO TO 4091
      GO TO (4083,4084,4085,4086,4087,4088), N
```

```
4083 U(K,J,I) = 0.0
     GO TO 4090
4084 W(K,J,I) = 0.0
     GO TO 4090
4085 U(K,J,I-1) = 0.0
     GO TO 4090
4086 W(K-1,J,I) = 0.0
     GO TO 4090
4087 V(K,J,I) = 0.0
     GO TO 4090
4088 V(K,J-1,I) = 0.0
4090 J1 = J1 + 1
4091 CONTINUE
     IF(ITEMP .EQ. 0) GO TO 4101
     IF(J1 .EQ. 0 .OR. J1 .EQ. 6) PRINT 4094, ICYCLE, J1, I, J, K
4094 FORMAT(//' ** COMMENT **  J1=0, THERE IS A CAVITY.  J1=6, FLOW FIE
    1LD MOVES TOO FAST(4101).' / ' VALUES OF ICYCLE, J1, I, J, K ARE '
    2  5I5)
4101 CONTINUE
4111 CONTINUE
C *CK IF A SUR CLL SHOULD BECOME A FUL CLL AND VICE VERSA.  *ID OF A SUR
C    CLL.
     DO 4361 K=2,KLMT
     DO 4361 J=2,JP1
     DO 4351 I=2,IP1
     M = IFLAG(K,J,I)/ICST3
     IF(M .LT. 3) GO TO 4351
     ICELL(I1)= IFLAG(K,J,I+1)/ICST3
     ICELL(I2)= IFLAG(K+1,J,I)/ICST3
     ICELL(I3)= IFLAG(K,J,I-1)/ICST3
     ICELL(I4)= IFLAG(K-1,J,I)/ICST3
     ICELL(I5)= IFLAG(K,J+1,I)/ICST3
     ICELL(I6)= IFLAG(K,J-1,I)/ICST3
     DO 4221 L=1,6
     IF(ICELL(L) .LT. 3) GO TO 4261
4221 CONTINUE
     GO TO (4351,4351,4231,4241,4231,4241,4351), M
4231 IFLAG(K,J,I) = IFUL
     GO TO 4351
4241 IFLAG(K,J,I) = IFH
     GO TO 4351
4261 GO TO (4351,4351,4271,4281,4271,4281,4351), M
4271 IFLAG(K,J,I) = IS
     GO TO 4291
4281 IFLAG(K,J,I) = ISH
4291 N = 0
     K1 = 1
     DO 4301 L=1,6
     IF(ICELL(L) .LT. 3) N=N+K1
4301 K1 = 2*K1
     IFLAG(K,J,I) = IFLAG(K,J,I) + N*ICST1
4351 CONTINUE
4361 CONTINUE
     RETURN
     END
'FOR,IS S04,S04
     SUBROUTINE VELCTY(IFLAG,PHI,U,V,W,D)
     COMMON/L1/ ITYPE, IBAR, JBAR, KBAR, IP2, JP2, KP2, IARRAY(5000),
```

```
      1  ARRAY1(5000), ARRAY2(5000), ARRAY3(5000), ARRAY4(5000),
      2 ARRAY5(5000)
       COMMON/L3/ IF, IFH, IS, ISH, IFUL, IFH, IB, ICST1, ICST3, IP1,
      1  JP1, KP1, ICYCLE, ICELL(6), KLMT
       COMMON/L4/ DXODY, DXODZ, DZODX, DZODY, DYODX, DYODZ
       DIMENSION IFLAG(KP2,JP2,IP2), PHI(KP2,JP2,IP2), U(KP2,JP2,IP2),
      1  V(KP2,JP2,IP2), W(KP2,JP2,IP2), D(KP2,JP2,IP2)
C #COMP THE VEL COMPS OF A SUR CLL AND OF EMP CLLS JUST OUTDIDE OF A FS.
       DO 4961 K=2,KLMT
       DO 4961 J=2,JP1
       DO 4961 I=2,IP1
       M = IFLAG(K,J,I)/ICST3
       IF(M .LT. 3 .OR. M .GT. 4) GO TO 4951
       L = (IFLAG(K,J,I) - M*ICST3)/ICST1
       IF(L .GT. 47) GO TO 4821
       IF(L .GT. 31) GO TO 4721
       IF(L .GT. 15) GO TO 4621
       N = L
       GO TO (4531,4541,4535,4545,4531,4551,4541,4561,4555,4541,4531,
      1  4565,4561,4545,4531), N
 4531  U(K,J,I) = U(K,J,I-1) - DXODY*(V(K,J,I) - V(K,J-1,I))
       U(K,J,I) = U(K,J,I) - DXODZ*(W(K,J,I) - W(K-1,J,I))
       GO TO 4951
 4535  U(K,J,I) = U(K,J,I-1)
 4541  W(K,J,I) = W(K-1,J,I) - DZODX*(U(K,J,I) - U(K,J,I-1))
       W(K,J,I) = W(K,J,I) - DZODY*(V(K,J,I) - V(K,J-1,I))
       GO TO 4951
 4545  U(K,J,I-1) = U(K,J,I) + DXODY*(V(K,J,I) - V(K,J-1,I))
       U(K,J,I-1) = U(K,J,I-1) + DXODZ*(W(K,J,I) - W(K-1,J,I))
       GO TO 4951
 4551  U(K,J,I-1) = U(K,J,I)
       GO TO 4541
 4555  U(K,J,I) = U(K,J,I-1)
 4561  W(K-1,J,I) = W(K,J,I) + DZODX*(U(K,J,I) - U(K,J,I-1))
       W(K-1,J,I) = W(K-1,J,I) + DZODY*(V(K,J,I) - V(K,J-1,I))
       GO TO 4951
 4565  U(K,J,I-1) = U(K,J,I)
       GO TO 4561
 4621  N = L - 15
       GO TO (4635,4631,4641,4645,4655,4635,4651,4641,4665,4645,4635,
      1  4631,4651,4665,4655,4635), N
 4631  U(K,J,I) = U(K,J,I-1)
 4635  V(K,J,I) = V(K,J-1,I) - DYODX*(U(K,J,I) - U(K,J,I-1))
       V(K,J,I) = V(K,J,I) - DYODZ*(W(K,J,I) - W(K-1,J,I))
       GO TO 4951
 4641  V(K,J,I) = V(K,J-1,I)
       GO TO 4541
 4645  U(K,J,I) = U(K,J,I-1)
 4651  V(K,J,I) = V(K,J-1,I)
       IF(L .EQ. 25) GO TO 4661
       IF(L .EQ. 29) GO TO 4655
       W(K,J,I) = W(K-1,J,I)
       IF(L .EQ. 19) GO TO 4951
 4655  U(K,J,I-1) = U(K,J,I)
       IF(L .EQ. 22) GO TO 4951
       IF(L .EQ. 20 .OR. L .EQ. 30) GO TO 4635
 4661  W(K-1,J,I) = W(K,J,I)
       GO TO 4951
```

```
4665 V(K,J,I) = V(K,J-1,I)
     GO TO 4561
4721 N = L - 31
     GO TO (4735,4731,4741,4745,4761,4735,4751,4741,4755,4745,4735,
    1  4731,4755,4755,4761,4735), N
4731 U(K,J,I) = U(K,J,I-1)
4735 V(K,J-1,I) = V(K,J,I) + DYODX*(U(K,J,I) - U(K,J,I-1))
     V(K,J-1,I) = V(K,J-1,I) + DYODZ*(W(K,J,I) - W(K-1,J,I))
     GO TO 4951
4741 V(K,J-1,I) = V(K,J,I)
     GO TO 4541
4745 U(K,J,I) = U(K,J,I-1)
     IF(L .EQ. 41) GO TO 4755
4751 W(K,J,I) = W(K-1,J,I)
4755 V(K,J-1,I) = V(K,J,I)
     IF(L .EQ. 41) GO TO 4661
     IF(L .EQ. 40 .OR. L .EQ. 45) GO TO 4561
     IF(L .EQ. 35) GO TO 4951
4761 U(K,J,I-1) = U(K,J,I)
     IF(L .EQ. 38) GO TO 4951
     IF(L .EQ. 44) GO TO 4661
     GO TO 4735
4821 N = L - 47
     GO TO (4635,4531,4541,4535,4545,4635,4551,4541,4561,4555,4541,
    1  4531,4565,4561,4545,4541), N
4951 CONTINUE
4961 CONTINUE
     K = 1
5101 K = K + 1
     J = 1
5105 J = J + 1
     I = 1
5111 I = I + 1
     M = IFLAG(K,J,I)/ICST3
     IF(M .LT. 3 .OR. M .GT. 4) GO TO 5451
     N = IFLAG(K,J,I+1)/ICST3
     IF(N .LT. 3 .OR. N .GT. 4) GO TO 5121
     LGC1 = (IFLAG(K+1,J,I)/ICST3)*ICST1 + IFLAG(K+1,J,I+1)/ICST3
     IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
    1            U(K+1,J,I)=U(K,J,I)-DZODX*(W(K,J,I+1)-W(K,J,I))
     LGC1 = (IFLAG(K-1,J,I)/ICST3)*ICST1 + IFLAG(K-1,J,I+1)/ICST3
     IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
    1            U(K-1,J,I)=U(K,J,I)+DZODX*(W(K-1,J,I+1)-W(K-1,J,I))
     LGC1 = (IFLAG(K,J+1,I)/ICST3)*ICST1 + IFLAG(K,J+1,I+1)/ICST3
     IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
    1            U(K,J+1,I)=U(K,J,I)-DYODX*(V(K,J,I+1)-V(K,J,I))
     LGC1 = (IFLAG(K,J-1,I)/ICST3)*ICST1 + IFLAG(K,J-1,I+1)/ICST3
     IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
    1            U(K,J-1,I)=U(K,J,I)+DYODX*(V(K,J-1,I+1)-V(K,J-1,I))
5121 N = IFLAG(K,J+1,I)/ICST3
     IF(N .LT. 3 .OR. N .GT. 4) GO TO 5221
     LGC1 = (IFLAG(K,J,I-1)/ICST3)*ICST1 + IFLAG(K,J+1,I-1)/ICST3
     IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
    1            V(K,J,I-1)=V(K,J,I)+DXODY*(U(K,J+1,I-1)-U(K,J,I-1))
     LGC1 = (IFLAG(K,J,I+1)/ICST3)*ICST1 + IFLAG(K,J+1,I+1)/ICST3
     IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
    1            V(K,J,I+1)=V(K,J,I)-DXODY*(U(K,J+1,I)-U(K,J,I))
     LGC1 = (IFLAG(K+1,J,I)/ICST3)*ICST1 + IFLAG(K+1,J+1,I)/ICST3
```

```
      IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
     1               V(K+1,J,I)=V(K,J,I)-DZODY*(W(K,J+1,I)-W(K,J,I))
      LGC1 = (IFLAG(K-1,J,I)/ICST3)*ICST1 + IFLAG(K-1,J+1,I)/ICST3
      IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
     1               V(K-1,J,I)=V(K,J,I)+DZODY*(W(K-1,J+1,I)-W(K-1,J,I))
 5221 N = IFLAG(K+1,J,I)/ICST3
      IF(N .LT. 3 .OR. N .GT. 4) GO TO 5451
      LGC1 = (IFLAG(K,J,I-1)/ICST3)*ICST1 + IFLAG(K+1,J,I-1)/ICST3
      IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
     1               W(K,J,I-1)=W(K,J,I)+DXODZ*(U(K+1,J,I-1)-U(K,J,I-1))
      LGC1 = (IFLAG(K,J,I+1)/ICST3)*ICST1 + IFLAG(K+1,J,I+1)/ICST3
      IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
     1               W(K,J,I+1)=W(K,J,I)-DXODZ*(U(K+1,J,I)-U(K,J,I))
      LGC1 = (IFLAG(K,J-1,I)/ICST3)*ICST1 + IFLAG(K+1,J-1,I)/ICST3
      IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
     1               W(K,J-1,I)=W(K,J,I)+DYODZ*(V(K+1,J-1,I)-V(K,J-1,I))
      LGC1 = (IFLAG(K,J+1,I)/ICST3)*ICST1 + IFLAG(K+1,J+1,I)/ICST3
      IF(LGC1.EQ.11 .OR. LGC1.EQ.12 .OR. LGC1.EQ.21 .OR. LGC1.EQ.22)
     1               W(K,J+1,I)=W(K,J,I)-DYODZ*(V(K+1,J,I)-V(K,J,I))
 5451 CONTINUE
      IF(I .LT. IP1) GO TO 5111
      IF(J .LT. JP1) GO TO 5105
      IF(K .LT. KLMT) GO TO 5101
      RETURN
      END
@FOR,IS S05,S05
      SUBROUTINE PLOT (I1,I2,I3,I4,M)
      COMMON/L2/ IX(100), IZ(100), X(2000), Y(2000), Z(2000), PLTX,
     1  PLTY, RPPUL, MARGNX, MARGNZ, NSEGPT, NIMRKR, COSPSI, SINPSI,
     2  CONTR1, CONTR2, CONTR3, CONTR4, XV(3,31), ZV(3,31)
      GO TO (5521,5721,5991), I1
 5521 IF(I4 .GT. 1) GO TO 5541
      DO 5531 I=1,9
      J = I + 1
 5531 CALL LINEV(IX(I),IZ(I),IX(J),IZ(J))
      DO 5535 I=11,NSEGPT,2
      J = I + 1
 5535 CALL LINEV(IX(I),IZ(I),IX(J),IZ(J))
      MCOUNT = 0
 5541 CONTINUE
      DO 5701 I=1,M
      MCOUNT = MCOUNT + 1
      XP = X(I)
      YP = Y(I)
      ZP = Z(I)
      IF(MCOUNT .GT. NIMRKR) GO TO 5601
      GO TO (5561,5591,5591,5591), I2
 5561 IF(YP .LT. PLTY .OR. XP .GT. PLTX) GO TO 5601
      GO TO 5701
 5591 CONTINUE
      GO TO 5701
 5601 IF(I3 .EQ. 1) GO TO 5681
      GO TO (5611,5641,5641,5641), I3
 5611 IF(XP .LT. CONTR4 .OR. XP .GT. CONTR1) GO TO 5701
      IF(YP .LT. CONTR4 .OR. YP .GT. CONTR2) GO TO 5701
      IF(ZP .LT. CONTR4 .OR. ZP .GT. CONTR3) GO TO 5701
      GO TO 5681
 5641 CONTINUE
```

```
 5681 JX = RPPUL*(XP + YP*COSPSI)
      JZ = RPPUL*(ZP + YP*SINPSI)
      JX = JX + MARGNX
      JZ = JZ + MARGNZ
      CALL PLOTV(JX,JZ,35,0)
 5701 CONTINUE
      RETURN
 5721 CONTINUE
      DO 5731 I=1,I4
      JX = XV(I3,I)
      JZ = ZV(I3,I)
      KX = XV(I3,I+1)
      KZ = ZV(I3,I+1)
 5731 CALL LINEV(JX,JZ,KX,KZ)
      DO 5741 I=1,M
      JX = X(I)
      JZ = Z(I)
      KX = X(I2+I)
      KZ = Z(I2+I)
      CALL LINEV(JX,JZ,KX,KZ)
      XP = KX - JX
      ZP = KZ - JZ
      J = SQRT(XP**2 + ZP**2)
 5741 IF(J .GT. 7) CALL ARROW(JX,JZ,KX,KZ,6,2)
 5991 CONTINUE
      RETURN
      END
'FOR,IS S21,S21
      SUBROUTINE CLKOUT(TCPU)
      CALL SCLOCK(DATE,TIME,ESEC,E60SEC)
      WRITE(6,1000) TIME
 1000 FORMAT( 6H0TIME= A12)
      CALL CPUTIM(ITIM)
      ESEC = FLOAT(ITIM)/1.E6
      WRITE(6,2000) ESEC
 2000 FORMAT( 13H0ESEC (CPU) =  F14.4)
      IF(ESEC .LT. TCPU) RETURN
      PRINT 3000
 3000 FORMAT(' ** MESSAGE **  EXECUTION IS TERMINATED TO PROTECT SC-4020
     1 PLOTS BEFORE MAX TIME IS REACHED. ')
      STOP
      END
'ASM,IL CPUTIM,CPUTIM          . USE AS     CALL CPUTIM(ITIM)
$(1)          AXR$             . WHERE ITIM IS ELAPSED CPU TIME
CPUTIM*       LA      A0,(23,ARRAY)      . IN MICROSECONDS
              ER      PCT$
              LA      A0,ARRAY+22
              MSI,XU  A0,200
              SA      A0,*0,X11
              J       2,X11
ARRAY         RES     23
              END
'FOR,IS S22,S22
      SUBROUTINE ARROW(IX1,IY1,IX2,IY2,IHITE,IBASE)
      HEIGHT = IHITE
      BASE = IBASE
      X2     = IX2
      Y2     = IY2
```

```
      DY     = IY2 - IY1
      DX     = IX2 - IX1
      SQ = 1./SQRT(DX*DX + DY*DY)
      FACTX = BASE * (SQ*DX)
      FACTY = BASE * (SQ*DY)
      X3 = X2 - HEIGHT * (SQ*DX)
      Y3 = Y2 - HEIGHT * (SQ*DY)
      IX4 = (X3 + FACTY) + .5
      IY4 = (Y3 - FACTX) + .5
      IX5 = (X3 - FACTY) + .5
      IY5 = (Y3 + FACTX) + .5
      CALL LINEV (IX4,IY4,IX2,IY2)
      CALL LINEV (IX5,IY5,IX2,IY2)
      RETURN
      END
@MAP,IL PROG,PROG
 LIB  SYS$*MSEC$.
 IN   SQ1
@COPOUT TPF$.,LHMAC3.
@REWIND,I LHMAC3.
@FREE LHMAC3.
@XQT  PROG
C
C   **** DATA DECK ****
C
@FIN
@FIN
```